Rapid Uncertainty Propagation and Chance-Constrained Path Planning for Small Unmanned Aerial Vehicles

Andrew W. Berning Jr.*[1] | Anouck Girard[2] | Ilya Kolmanovsky[3] | Sarah N. D'Souza[4]

[1]Research Assistant, Department of Aerospace Engineering, University of Michigan, MI, USA
[2]Associate Professor, Department of Aerospace Engineering, University of Michigan, MI, USA
[3]Professor, Department of Aerospace Engineering, University of Michigan, MI, USA
[4]Aerospace Flight Systems Engineer, NASA Ames Research Center, CA, USA

Correspondence
Email: awbe@umich.edu

Summary

With the number of small Unmanned Aircraft Systems (sUAS) in the national airspace projected to increase in the next few years, there is growing interest in a traffic management system capable of handling the demands of this aviation sector. It is expected that such a system will involve trajectory prediction, uncertainty propagation, and path planning algorithms. In this work, we use linear covariance propagation in combination with a quadratic programming-based collision detection algorithm to rapidly validate declared flight plans. Additionally, these algorithms are combined with a Dynamic, Informed RRT* algorithm, resulting in a computationally efficient algorithm for chance-constrained path planning. Detailed numerical examples for both fixed-wing and quadrotor sUAS models are presented.

KEYWORDS:
Stochastic Optimization, Path Planning, Uncertainty Quantification

1 | INTRODUCTION

1.1 | Motivation

There were 110,604 registered small Unmanned Aircraft Systems (sUAS) in the United States at the end of 2017, and that number is expected to quadruple by 2022[1]. There has been great interest, accordingly, in a UAS Traffic Management (UTM) system to handle the demands of this rapidly growing aviation sector[2,3]. In at least one potential design of such a system, real-time communication channels exist between a central computation platform and the sUAS. This would provide the vehicles access to valuable computation resources and information about nearby vehicles, terrain, and atmospheric conditions, allowing for safe and efficient route planning.

One of the responsibilities of such a UTM system will likely be to apply a risk-based approach where geographical needs and use cases determine the airspace performance requirements[4]. Addressing this will require a sUAS trajectory prediction model that validates vehicle flight performance and allows for UTM to determine whether or not the vehicle can operate in the airspace given real-time information about wind, other vehicles, and/or obstacles. Given a large number of vehicles predicted to be in operation, this trajectory prediction model must accommodate multiple vehicle types and airspace environments (wind, terrain, etc.). Another challenge in this setting is that the trajectory could depend on proprietary information such as control systems, methods, and gain-tuning specific to a particular vehicle. Implementing a system that is reliant on such information could be prohibitive due to the constant modifications that would be required to accommodate diverse and evolving control laws, and due to the potential legal barriers to acquiring proprietary information from all operators.

In the context of UTM operations, where varied vehicle types and uncertainties are expected, there is a need for a model that represents expected vehicle performance while accounting for the uncertainty in the context of operational environments. The

motivation for the work detailed in this paper is to develop computationally efficient trajectory validation and planning algorithms that utilize uncertainty quantification (UQ) and propagation techniques to provide an assessment of the vehicle performance and risk of violating constraints.

1.2 | Problem Statement

The work described in this paper focuses on two separate but related problems: rapid uncertainty propagation for trajectory validation and chance-constrained path planning.

For the former, we seek an algorithm that takes the vehicle's dynamics, initial state, parameters, and desired trajectory as inputs, and outputs some quantification of the uncertainty associated with the vehicle's state trajectory over the specified flight horizon, as well as an assessment of whether the probability of trajectories violating constraints is sufficiently low at any given time instant. For the latter, we are interested in the ability to re-plan or repair the trajectory if it is found that the vehicle's probabilistic trajectory tube intersects with a keep-out zone. Such a tube and constraint are illustrated in Figure 1.
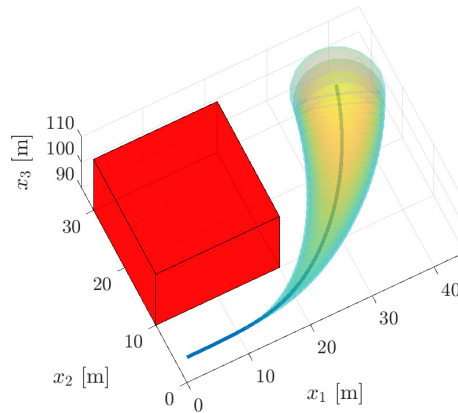

FIGURE 1 Nominal trajectory (dotted), probabilistic trajectory tube in the presence of uncertainties and an obstacle (red).

1.3 | Literature Review

The source of uncertainty considered in this work is atmospheric turbulence and stochastic wind gusts, which affect the response[5] and safety[6,7] of aircraft. These effects are particularly pronounced in relatively lightweight sUAS. Hoblit[8] provides a detailed description of discrete and continuous gust models and Richardson[9] provides a thorough review of the modeling techniques involved. The Dryden and Von Kármán models are the two most common gust/turbulence models[10,11,12,13,14], which are used in both Federal Aviation Administration (FAA) and military specifications. Both consider the linear and angular velocity components of the gusts to be varying stochastic processes and then specify those components' Power Spectral Density (PSD). The Dryden model utilizes rational PSDs, while the Von Kármán model uses irrational PSDs. The latter matches experimental gust observations more closely than the former, but its use of irrational spectral densities prevents its spectral factorization from being exactly expressed.

We also consider the problem of propagating the vehicle's state uncertainty subject to the aforementioned wind gust disturbances. Given an initial probability distribution of a state, the objective of a UQ algorithm is to obtain a characterization of the state's probability distribution at a future instant in time. For this work, we require the algorithm to compute the uncertainty for all time instants between the initial and final times. The conceptually simple solution is through a Monte Carlo (MC) simulation, but the computational intensity of this method limits its usefulness for our application[15,16]. Linearization techniques suffer from

diminished accuracy for highly nonlinear systems or for long time horizons, but their simplicity and high computational efficiency make them well-suited for real-time trajectory optimization or path planning [17,18]. Other nonlinear UQ methods include unscented transformation (UT) [19,20], polynomial chaos (PC) expansions [21], and Gaussian mixture models (GMM) [22,23,24]. A thorough survey of many other UQ methods is provided by Luo [25], though none of these were deemed appropriate for our use due to our models' large number of states, the presence of stochastic inputs, and the need for rapid computations. For that reason, the linear covariance propagation method is an attractive choice for our problem. The ease of incorporating exogenous disturbances and computing a complete time history of the covariance further distinguish it from the other UQ methods.

There is a wide array of solution strategies for planning a vehicle's path subject to uncertainty, including convex programming [26], mixed integer linear programming [27], graph search [28], fast marching trees [29], and probabilistic roadmaps [30]. One of the more popular approaches [31,32,33,34,35] utilizes a class of stochastic search algorithms called Rapidly-exploring Random Trees (RRTs) [36]. These algorithms are well suited for real-time implementation [32] and are sampling-based, so they scale well with problem size, but only offer a probabilistic guarantee of completeness. A comparison of sequential quadratic programming, genetic algorithms, and RRT is provided by Borowski and Frew [35]. Extensions to RRT such as RRT* [37], Informed RRT* [38], and Dynamic RRT* [39], improve optimality of the solution, increase sampling efficiency, and allow for dynamically changing constraints, respectively. These three RRT extensions are utilized in this work.

1.4 | Original Contributions

The main contribution of this paper is the amalgamation of existing Dynamic RRT* and Informed RRT* algorithms, and the addition of an obstacle buffer resizing technique, resulting in a single algorithm that allows computationally efficient, chance-constrained path planning for sUAS. In particular, this contribution addresses a dilemma in chance-constrained path planning under uncertainty: trajectory re-planning changes the outcome of the covariance propagation, which, when obstacles or exclusion zones are involved, may require further re-planning.

This paper builds upon previous work [40] that considered rapid uncertainty propagation, collision detection, and trajectory optimization for fixed-wing, 2D longitudinal flight dynamics. The numerical examples presented here utilize both fixed-wing and quadrotor sUAS in full 3D flight in a non-static atmosphere with inner loop and outer loop controllers in closed form. Other contributions in support of the path planning algorithm are linear covariance propagation of the uncertainty in initial states and exogenous disturbances for three dimensional vehicle motion, and a quadratic programming-based approach to 3D collision detection.

2 | MODELING

The purpose of this paper is to present and illustrate with simulations a procedure for rapid uncertainty propagation and chance constrained path planning that is broadly applicable in the sUAS setting in terms of models and uncertainties assumed. This section introduces the two systems used for numerical examples in this work: a quadrotor sUAS model and a fixed-wing sUAS model.

The quadrotor sUAS is modeled as a double integrator with quadratic aerodynamic drag, constant drag coefficient, dynamic extension controller, and a Dryden-based wind disturbance. The full model can be found in Appendix A. The fixed-wing sUAS model is based on a 6-state model for longitudinal and lateral flight in a moving atmosphere, with inner and outer-loop controllers added and a Dryden wind gust model. The full model can be found in Appendix B.

The above models given by Eqs. (A2) – (A11) for the quadrotor or Eqs. (B38) – (B43) for the fixed-wing aircraft can be viewed as a single system of differential equations of the form:

$$\dot{X} = f\left(X, X_{des}(t), n(t), \theta\right), \tag{1}$$

where the state is $X = [r, V_0, \eta_{x1}, \eta_{x2}, \eta_{x3}]^\top$ for the quadrotor model or $X = [x, y, h, V, \psi, \gamma, T, V_{des}, \psi_{des}, \eta_u, \eta_w, \eta_v]^\top$ for the fixed-wing model, θ is the set of vehicle, environmental, and control parameters, and n is the white noise input.

3 | TECHNICAL APPROACH

3.1 | Covariance Propagation

For the uncertainty propagation analysis, we linearize the sUAS model in Eq. (1). The linearized model has the following form:

$$\delta \dot{X} = A(t)\delta X(t) + B_n(t)n(t), \tag{2}$$

$$A(t) = \frac{\partial f}{\partial X}\bigg|_{\substack{X=\bar{X}(t) \\ n=0 \\ X_{des}(t)}}, \tag{3}$$

$$B_n(t) = \frac{\partial f}{\partial n}\bigg|_{\substack{X=\bar{X}(t) \\ n=0 \\ X_{des}(t)}}. \tag{4}$$

Here, $\bar{X}(t)$ is the nominal trajectory obtained by propagating Eq. (1) subject to $n = 0$ and specified time history of X_{des}. Physically, this corresponds to the nominal trajectory that the sUAS would fly under no-wind conditions.

Under the assumption that $n(t)$ is a standard white noise process, the state covariance matrix, P, satisfies the Lyapunov differential equation[41]:

$$\dot{P}(t) = A(t)P(t) + P(t)A^\top(t) + B_n(t)B_n^\top(t). \tag{5}$$

Note that (5) can be solved using any standard ODE solver. The initial condition for P is the zero matrix if the vehicle's initial state is known deterministically, or may be chosen as a diagonal matrix of state covariances, otherwise.

Now, we can use the nominal trajectory $\bar{X}(t)$, covariance matrix $P(t)$, and a Gaussian distribution density function to estimate the probability that the vehicle is contained in a specified set at time t. For the quadrotor model: given the mean $\bar{r}(t) = [\bar{x}_1(t), \bar{x}_2(t), \bar{x}_3(t)]^\top$ and the block of the covariance matrix corresponding to these states, $\Sigma_\omega(t) = P_{(1:3,1:3)}(t)$, we can build a set from the definition of a probability ellipsoid[42]:

$$Prob\left[\omega \in \mathbb{R}^3 : (\omega - \bar{r}(t))^\top \Sigma_\omega^{-1}(t)(\omega - \bar{r}(t)) \leq c^2\right] = \beta, \tag{6}$$

where $\beta \in [0, 1]$ is a prescribed probability level and c is solved for using the three degree-of-freedom chi-squared distribution[43]. Thus the vehicle has a β probability of being within the set (6) at time t.

For verification purposes, Eq. (1) is also solved numerically in a series of MC simulations. For these computations, the white Gaussian noise is computed using MATLAB's *randn*() function, scaled by the inverse square root of the integration time step[44]. A comparison between the linear covariance propagation and the covariance matrix estimated from the MC runs is shown in Section 4.

3.2 | Collision Detection

For trajectory validation purposes, it is desirable to have a collision detection algorithm that can quickly determine whether or not the trajectory tube formed by the set of probability ellipsoids defined in Eqn. (6) intersects with an obstacle, represented in this work as a cuboid, as illustrated in Figure 2.

The ellipsoid-cuboid intersection at time instant t is formulated as a quadratic programming (QP) problem that can be solved by a standard QP solver:

$$\min_z \; (z - \bar{r}(t))^\top \Sigma_\omega^{-1}(t)(z - \bar{r}(t)) \tag{7}$$

$$s.t.$$

$$A_O z \leq b_O.$$

Here, z, the optimization parameter, is a concatenation of x_1, x_2 and x_3 coordinates: $z = [x_1, x_2, x_3]^\top$ and $A_O z \leq b_O$ are the linear inequalities that represent the cuboid obstacle. Thus we are solving for the point z that belongs to the smallest ellipsoid centered at $\bar{r}(t)$ that still touches the cuboid obstacle defined by $A_O z \leq b_O$. Let z^* denote the solution to this QP and let $c^{*2}(t) = (z^* - \bar{r}(t))^\top \Sigma_\omega^{-1}(t)(z^* - \bar{r}(t))$. If $c^{*2}(t) \geq c^2$, then there is no intersection between the ellipsoid and cuboid at time t, and the chance constraint is not violated.
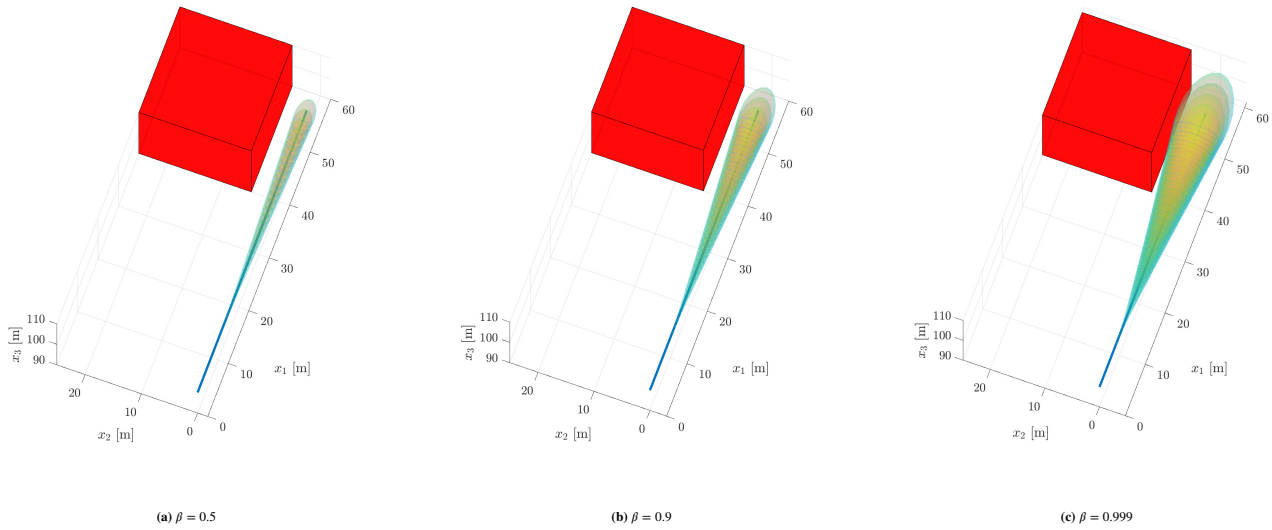
(a) $\beta = 0.5$ (b) $\beta = 0.9$ (c) $\beta = 0.999$

FIGURE 2 Collision scenario for varying values of β.

For the numerical implementation of this collision detection scheme, the frequency at which collisions are checked is an algorithm tuning parameter, and (7) is only solved after it has been determined that the overbounding spheres of the ellipsoid and cuboid intersect. The latter check is computationaly very simple.

3.3 | Path Planning

In our prototypical UTM scenario mentioned in Section 1, the covariance propagation and tube generation algorithms discussed in Section 3.1 inform the functionality to validate aircraft trajectories. If a desired flight plan is found to be unsafe due to possible collisions with obstacles, the UTM system should return a safe flight path for the UAS to fly. Here, the UAS is operating in a densely populated urban environment with many no-fly zones or obstacles. For scenarios such as this, with a large number of nonconvex obstacle constraints, a Rapidly-expanding Random Tree (RRT) algorithm is useful.

This section lays out the modifications made to the standard 2D RRT algorithm[36] for the purpose of generating a desired constant-altitude flight path X_{des}. Note that the path planning is handled in two dimensions, under the assumption that the UAS operates in an urban environment at a constant desired altitude, but all tube generation and collision checks are three-dimensional.

3.3.1 | Informed Subset

By their nature, RRT* algorithms find optimal paths between the initial state and every state in the search space, regardless of what final state the user is actually interested in. This is an inefficient use of computational resources that the Informed variant[38] attempts to address. By only sampling within an ellipsoidal subset and excluding regions of the sample space that cannot possibly improve the solution, the algorithm makes a better-informed decision about where to look for optimal paths.

This informed subset is defined as follows:

$$Q_{\hat{f}} = \left\{ q \in Q \ \middle| \ \|x_{start} - q\| + \|q - x_{goal}\| \le c_{best} \right\} \tag{8}$$

where c_{best} is the current best solution cost, Q is the global sample space, x_{start} is the initial state and x_{goal} is the desired final state and $\|\cdot\|$ denotes the euclidean distance.

This Informed variant is outlined in Algorithm 1 from[38], where $\mathcal{T}$ is the current set of nodes, r_w is the rewiring radius, and N is the total number of iterations. Note that the subroutine $SampleEllipse(x_{start}, x_{goal}, c_{best})$ returns a randomly sampled point that is contained within the ellipse defined in (8), and lines 6–17 in Algorithm 2 include the rewiring steps essential to the RRT* algorithm variant.

Algorithm 1 $\mathcal{T} = $ Informed RRT$^*(x_0, x_f)$

1: $\mathcal{T}_0 \leftarrow x_0$
2: $c_{best} \leftarrow \infty$
3: **for** $k = 1 \dots N$ **do**
4:     $\mathcal{T}_{k+1} = AddNode(\mathcal{T}_k)$
5: **end for**
6: **return** $\mathcal{T}_N$

Algorithm 2 $\mathcal{T}_{k+1} = AddNode(\mathcal{T}_k)$

1: $x_{rand} \leftarrow SampleEllipse(x_{start}, x_{goal}, c_{best})$; % c_{best} is cost of current best solution
2: $x_{nearest} \leftarrow Nearest(\mathcal{T}, x_{rand})$;
3: $x_{new} \leftarrow Steer(x_{nearest}, x_{rand})$;
4: **if** $NoCollision(x_{nearest}, x_{new})$ **then**
5:     $\mathcal{T} \leftarrow \mathcal{T} + x_{new}$;
6:     $X_{near} \leftarrow Near(\mathcal{T}, x_{new}, r_w)$;
7:     $x_{min} \leftarrow x_{nearest}$;
8:     $c_{min} \leftarrow Cost(x_{min}) + ||coordinates(x_{nearest}) - coordinates(x_{new})||$
9:     **for** $\forall x_{near} \in X_{nearest}$ **do**
10:         $c_{new} \leftarrow Cost(x_{near}) + ||coordinates(x_{near}) - coordinates(x_{new})||$
11:         **if** $c_{new} < c_{min}$ **then**
12:             **if** $NoCollision(x_{nearest}, x_{new})$ **then**
13:                 $x_{min} \leftarrow x_{near}$
14:                 $c_{min} \leftarrow c_{new}$
15:             **end if**
16:         **end if**
17:     **end for**
18:     $Parent(x_{new}) \leftarrow x_{min}$
19: **end if**
20: **return** $\mathcal{T}$

3.3.2 | Chance Constraints and Dynamic Extension

If the solution of Algorithm 1 was passed to the tube generation algorithm as X_{des}, there would be no guarantee that the generated tube would not intersect the obstacles. The solution to a path planning problem with obstacles often passes very closely to the obstacle, where any nonzero uncertainty in the state would produce a tube that would intersect the obstacle. Thus a buffer region is added to the obstacles and adjusted periodically in a manner informed by full covariance propagation.

This buffer can be seen in Figure 3, which represents a single outer-loop iteration in which the buffer is held constant. Here, the green triangle is the starting point x_{start}, the blue circle is the end point x_{goal}, the green line is the current best solution, the dashed red ellipse is the current informed subset as defined in Eqn. (8), and the black lines represent the total set of sampled nodes and connecting vertices. The solid red rectangle is the actual obstacle, while the dashed red rectangle is the buffered obstacle that is actually passed to Algorithm 1 and used in the subroutine $NoCollision(x_{nearest}, x_{new})$. The bottom subfigure is the same as the top, but with the covariance tube as defined by (6) overlaid, illustrating how the buffer prevents the tube from intersecting the true obstacle.

The dynamic extension to this algorithm involves intermittently halting the standard RRT algorithm, propagating the nominal vehicle trajectory and covariance, and adjusting the obstacle buffer sizes accordingly. This requires an algorithm that informs the buffer size adjustments, answering the question: "By how much should we grow or shrink the buffer such that the covariance tube just touches the obstacle?". To answer this, we start with the easier question of "What size ellipsoid just touches the obstacle?" This is illustrated in Figure 4 and can be expressed as a QP:
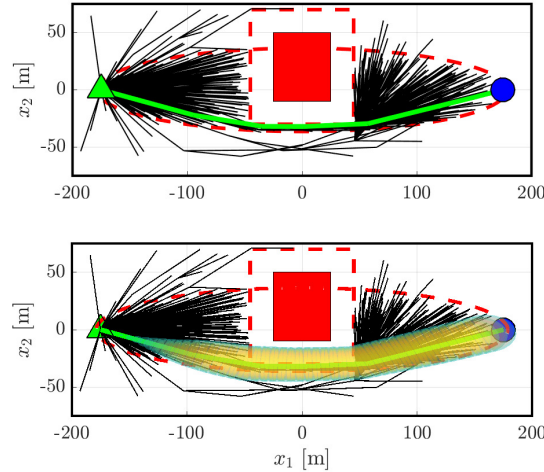
FIGURE 3 Results of Algorithm 1 with a buffered obstacle.

$$\min_{z}(z - \bar{r}(t))^{\top}\Sigma_r^{-1}(t)(z - \bar{r}(t)) \tag{9}$$

$$s.t.$$

$$A_O z \le b_O. \tag{10}$$

We alter (10) to take into account the size of the buffer as follows:

$$A_O z \le b_O + ld \tag{11}$$

where l is a vector of ones and d is a scalar representing the size of the buffer. We can then vary d until $c^{2*} = c^2$, which solves for the buffer size such that the smallest ellipsoid that intersects the buffer is the same size as the actual covariance ellipsoid. This solution is used as the buffer size for the next iteration. Note that this new buffer size is only an approximation of the actual solution since the size of the covariance tube is dependent on the solution to the inner loop, X_{des}, which changes every iteration.

Letting l be a vector of all ones has the effect of buffering the obstacle by an equal amount in all directions, as opposed to extending the buffer only in certain directions. Equal buffering on all sides of the obstacle may result in less optimal trajectories in a situation where the vehicle's trajectory interacts with multiple edges of an obstacle and this is viewed as the trade-off for not including the computationally nontrivial logic for a more discriminating buffer re-sizing. Additionally, expanding the buffer in all directions reduces the RRT search space more than merely expanding in one direction, which aids convergence.

Because the size of the probability ellipsoids vary spatially, the optimal buffer size is not necessarily the same for every obstacle. Thus, buffer changes happen independently for every obstacle. There are two cases to be considered: the buffer shrinks or stays the same, and the buffer grows. If a buffer is shrunken or left unchanged, no RRT nodes are changed, and the algorithm continues as normal. However, if a buffer grows, the nodes and connecting vertices that are now contained within the newly-sized obstacle are no longer valid. They are thus deleted, leaving only the parent trees and the now-stranded branches. Then, the Informed RRT* algorithm is used to grow branches from the parent tree until any part of the stranded branch is re-connected. At this point, any nodes left without a parent are removed from the set of nodes, and the Informed RRT* algorithm continues to run according to Algorithm 1 until the next covariance propagation and buffer resize. This is laid out in more detail in Algorithm 3.

4 | RESULTS

4.1 | Trajectory Validation

The first result of interest is a validation of the linear covariance (LC) propagation described in Section 3.1 versus the results of the Monte Carlo simulations. For both the fixed-wing and quadrotor models, a simple desired trajectory was prescribed, and
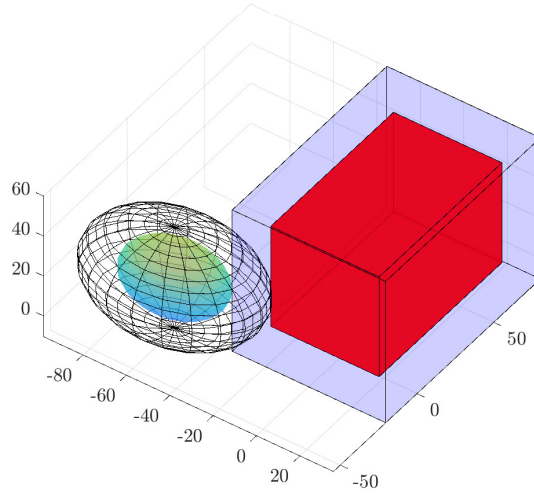
FIGURE 4 Cuboid buffer expansion illustration. The shaded ellipsoid and red cuboid represent the original confidence ellipsoid and obstacle, respectively. The wire mesh ellipsoid and blue cuboid represent the expanded sets that just touch the original obstacle and confidence ellipsoid, respectively.

the state covariance computed from a 10,000 run MC simulation is compared to the LC results computed from Eqn. (5). For the quadrotor model, the desired trajectory is a three phase ascent-cruise-descent mission profile. For the fixed-wing model, it is a constant-altitude trajectory with a sinusoid shape in the lateral direction.

The comparison results are shown in Figures 5 − 8. Here, $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$, $\bar{x}$, $\bar{y}$, and $\bar{h}$ represent either the mean position, or the nominal trajectory $\bar{X}(t)$, from the MC and LC simulations, respectively. All plots show good agreement between MC and LC, with the largest deviations being $\approx 15\%$ in the σ_{x2}^2 channel in Figure 5. This justifies our use of LC propagation in path planning under uncertainty. Note that the deviations in $\bar{x}_2$ appear large in the plot, but the vertical axis range is quite small. Also note that the nominal trajectory for the fixed-wing example does not track the desired trajectory exactly, but that is a function of the model's controller, which is not the focus of this work, as explained in Section 2.

All computations are performed in the MATLAB environment running on a dual-core Intel Core i5 at 2.7 GHz, and all confidence ellipsoid tubes use $\beta = 0.999$. This value was chosen for illustration purposes, though some UTM applications may call for smaller or larger values. For the quadrotor model, the LC propagation took $0.062s$ of computation time for $40s$ of simulated flight time. For the fixed-wing model, it took $0.18s$ of computation time for $35s$ of simulated flight time.
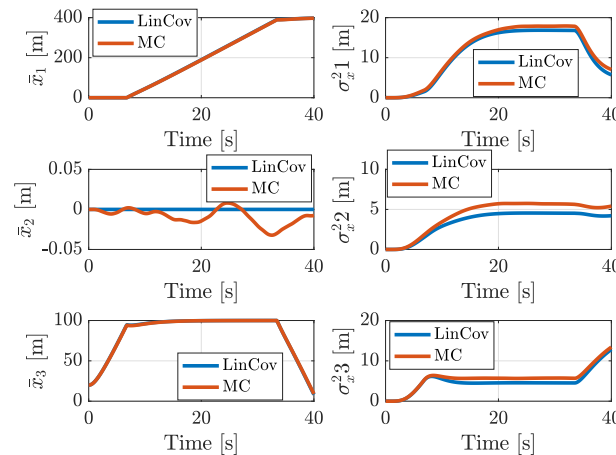

FIGURE 5 State and covariance time histories for prescribed quadrotor trajectory.

Algorithm 3 $\mathcal{T}$ = Dynamic Informed RRT*(x_0, x_f)

1: $\mathcal{T}_0 \leftarrow x_0$
2: $c_{best} \leftarrow \infty$
3: $\delta \leftarrow \infty$
4: **for** $m = 1 \dots M$ **do**
5:      $k = 0$
6:      **while** $k < N_{max}$ AND $\delta > tol$ **do**
7:          $k = k + 1$
8:          $\mathcal{T}_k = AddNode(\mathcal{T}_{k-1})$
9:          $cost_i = FindBestPath(\mathcal{T}_i)$
10:         **if** $i > N_{conv}$ **then**
11:             $\delta = |\frac{cost_i - cost_{i-N_{conv}}}{cost_{i-N_{conv}}}|$
12:         **end if**
13:     **end while**
14:     **if** $m \neq M$ **then**
15:         $d = CompObsDist(\mathcal{T}_k, O)$ % Compute distance between obstacle and covariance tube
16:         **for** $j = 1 \dots N_{obs}$ **do**
17:             **if** $d_j \geq 0$ **then**
18:                 $b_j = b_j - d_j$
19:             **else if** $d_j < 0$ **then**
20:                 $b_j = b_j - d_j$
21:                 $\mathcal{T} = CleanupNodes(\mathcal{T}, O_j)$ % remove nodes that are no longer valid
22:                 $\mathcal{T} = Regrow(\mathcal{T}, O_j, params)$ % regrow until orphaned nodes are reconnected or pruned
23:             **end if**
24:         **end for**
25:     **end if**
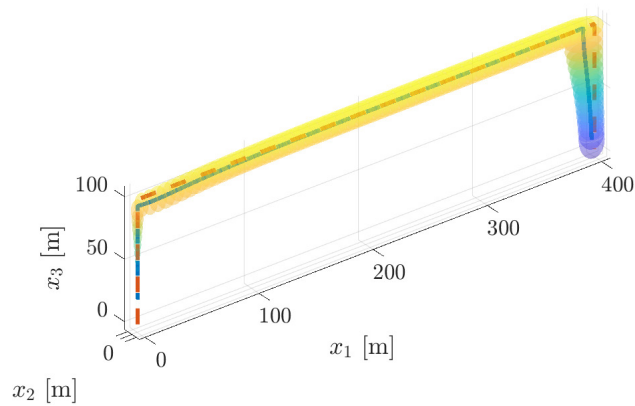26: **end for**
27: **return** $\mathcal{T}$

FIGURE 6 Quadrotor 3D trajectory. The covariance tube is constructed from a set of confidence ellipsoids defined by (6), the dashed line is the desired trajectory, and the solid line is the nominal trajectory $\bar{X}(t)$.
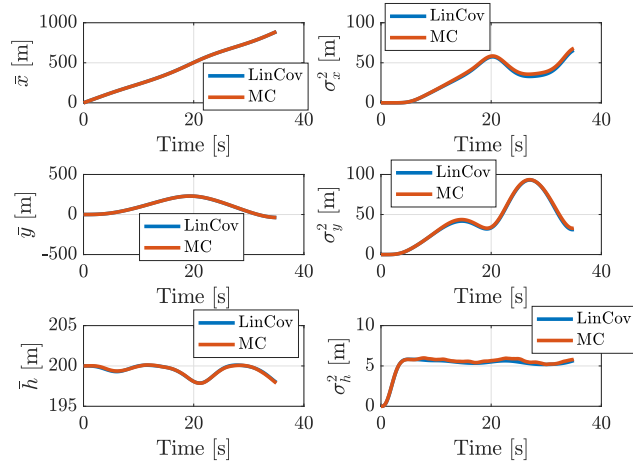
FIGURE 7 State and covariance time histories for prescribed fixed-wing trajectory.
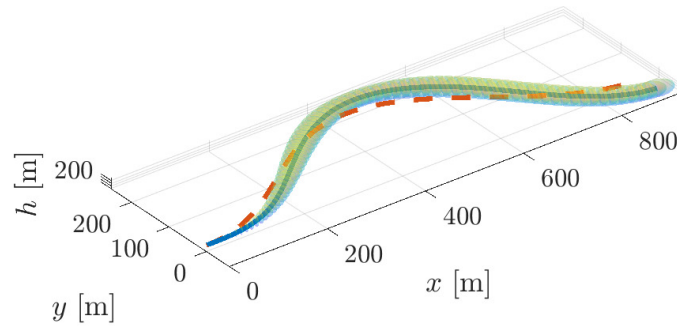

FIGURE 8 Fixed-wing sUAS 3D trajectory. The covariance tube is constructed from a set of confidence ellipsoids defined by (6), the dashed line is the desired trajectory, and the solid line is the nominal trajectory $\bar{X}(t)$.

4.2 | Path Planning

Now that we have shown that the LC propagation scheme is sufficiently fast and accurate when compared to MC simulations of the nonlinear system, we can use the LC propagation algorithm to enable chance-constrained path planning as laid out in Section 3.3. For brevity, only the quadrotor model is utilized in this section.

Figures 9 and 10 show the results of Algorithm 3 for a three obstacle scenario. A solution is found that is very close to the minimum path length between x_{start} and x_{goal}, while still ensuring that the β confidence ellipsoid tube does not intersect any of the obstacles. Total computation time for this example was $30.5s$.

Note that the buffer re-sizing algorithm converges, but only to a region, the size of which is dictated by *tol*. There is no guarantee that the algorithm will exit without a constraint violation, but, based on extensive simulation studies, the magnitude of the violation will be sufficiently small for sufficiently large M and small *tol*.

5 | CONCLUSIONS

Future UTM systems will need to rely on rapid UQ for trajectory validation and path planning in order to account for the effects of wind turbulence/gusts and other uncertainties and disturbances. This work combines existing Dynamic RRT* and Informed RRT* algorithms, and adds an obstacle buffer resizing technique to solve a challenge of chance-constrained path planning: trajectory re-planning changes the outcome of the covariance propagation. The results presented here shows that this RRT*-based
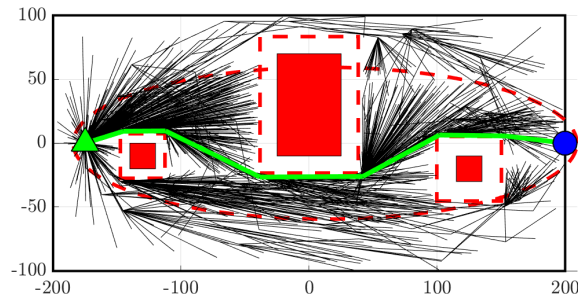
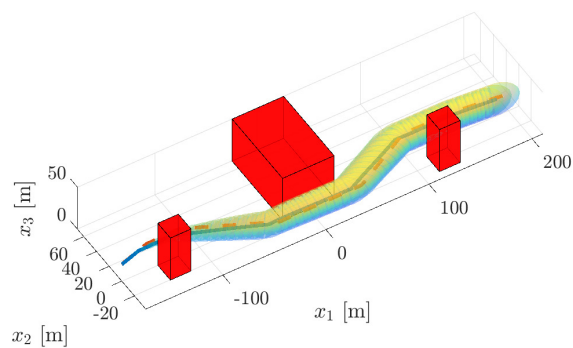**FIGURE 9** Chance-constrained Informed Dynamic RRT* solution for quadrotor model with three obstacles.

FIGURE 10 3D plot of obstacles and flight path with β confidence tube overlaid.

algorithm, in combination with QP-based collision checking for trajectory validation, successfully solves the aforementioned problem, resulting in a computationally efficient chance-constrained path planner. Trajectory validation examples were presented for both quadrotor and fixed-wing models in 3D flight in a non-static atmosphere. Path planning examples were presented for the quadrotor model, showing near-optimal navigation around 3 obstacles while enforcing chance state constraints.

Financial disclosure

Conflict of interest

The authors declare no potential conflict of interests.

☐

APPENDIX

A QUADROTOR MODELING

As a balance between model fidelity and computational efficiency, the quadrotor model used for uncertainty propagation is built upon double integrator dynamics augmented with aerodynamic drag of the following form:

$$r = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \; V_0 = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}, \; V_q = V_0 - w, \tag{A1}$$

$$\dot{V}_0 = u - \frac{1}{2m} \rho S C_D V_q \left\| V_q \right\|, \tag{A2}$$

where x_1, x_2, and x_3 are the vehicle's coordinates in an inertial reference frame, V_q is its velocity with respect to the atmosphere, $w \in \mathbb{R}^3$ is the velocity of the local atmosphere with respect to the ground, $u \in \mathbb{R}^3$ is the control input, m is the vehicle mass, ρ is the air density, S is the reference area, and C_D is the coefficient of drag, assumed to be constant. This model assumes that an inner-loop controller for vehicle attitude and thrust has been implemented that has a feed forward term to cancel out gravity and has sufficiently high bandwidth that we can control acceleration directly. It also assumes a non-static atmosphere and a drag coefficient that remains constant regardless of vehicle state or direction of the relative wind vector V_q. Uncertainty in the drag coefficient can, however, be handled using our UQ and planning algorithms.

When using the above model for uncertainty propagation, it is important that a controller be added so that the uncertainty in vehicle states does not grow so large as to be useless for trajectory prediction purposes. The method presented here is agnostic to controller choice, but for modeling purposes, it is beneficial if the controller can be expressed in a closed form that is easily linearizable. This work utilizes a dynamic extension controller of the following form:

$$e = r - r_{des}, \tag{A3}$$

$$S_c = \dot{e} + K_q e, \tag{A4}$$

$$u = \ddot{r}_{des} - K_q \dot{e} - \Lambda_q S_c, \tag{A5}$$

where $K_q \in \mathbb{R}^{3\times3}$ and $\Lambda_q \in \mathbb{R}^{3\times3}$ are controller gains that may be treated as tuning parameters to match the flight characteristics of our model to the flight characteristics of a real-world vehicle for which the UTM may be trying to predict its trajectory.

Finally, we consider the model for our wind disturbance. A Dryden wind model specifies the power spectral density (PSD) for the body-fixed longitudinal, lateral, and vertical directions of a fixed-wing aircraft. Other works [45] apply this model directly to quadrotors. Here, the longitudinal channel is replicated in the x_1, x_2, and x_3 directions of the inertial frame. The resulting Dryden-like wind model is summarized as

$$H_i(s) = \sigma_i \sqrt{\frac{2L_i}{\pi \|V_0\|}} \frac{1}{1 + \frac{L_i}{\|V_0\|} s}, \tag{A6}$$

$$A_i = \frac{-\|V_0\|}{L_i}, \tag{A7}$$

$$B_i = 1, \tag{A8}$$

$$C_i = \sqrt{2} \|V_0\| \sigma_i \sqrt{\frac{L_i}{\|V_0\|}} \frac{1}{L_i \sqrt{\pi}}, \tag{A9}$$

$$\dot{\eta}_i = A_i \eta_i + B_i n, \tag{A10}$$

$$w_i = C_i \eta_i, \tag{A11}$$

for $i = \{x_1, x_2, x_3\}$. Here, H_i is the transfer function of Dryden model coloring filter, A_i, B_i, and C_i are its state space realization, σ_i is the gust intensity parameter, L_i is the characteristic length, and n is the Gaussian white noise input [8]. The output of this model, $w = [w_1, w_2, w_3]^\top$, is the wind velocity vector in (A2).

B FIXED-WING AIRCRAFT MODELING

To illustrate another typical setting which involves sUAS, the EOMs for a fixed-wing aircraft under closed-loop control in a non-static atmosphere were derived. Hull[46] provides EOM's for longitudinal flight with moving atmosphere and EOM's for full three-dimensional flight in a static atmosphere, but does not provide EOM's for 3D flight in a moving atmosphere.

Figure B1 shows the unit vectors and rotations necessary to define flight in three dimensions. Here, frame A is the inertial frame, while frames B, C, and D are rotated by angles ψ, γ, and μ, respectively, where ψ is velocity yaw or heading angle, γ is velocity pitch, and μ is velocity roll.
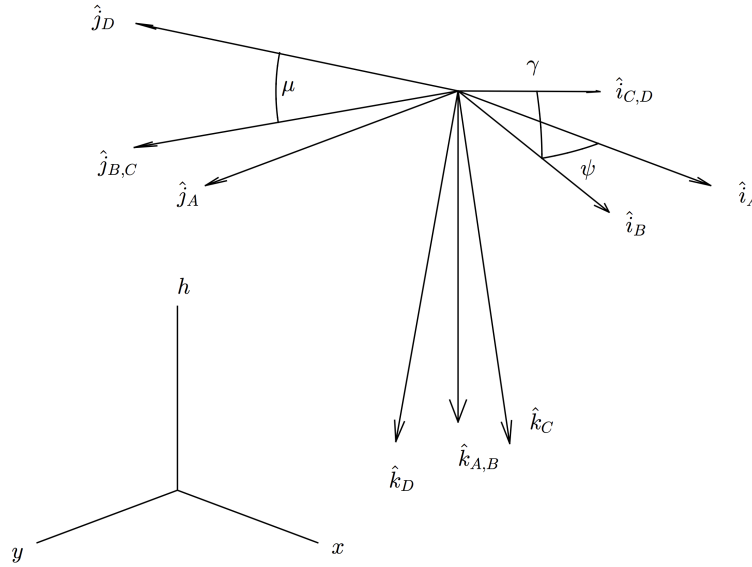

FIGURE B1 Three-dimensional flight: unit vectors and rotations.

Additionally, the thrust, drag, lift and gravity forces acting on the aircraft can be defined as follows:

$$\vec{T} = T\hat{i}_D, \tag{B12}$$

$$\vec{D} = -D\hat{i}_D, \tag{B13}$$

$$\vec{L} = -L\hat{k}_D, \tag{B14}$$

$$m\vec{g} = mg\hat{k}_A, \tag{B15}$$

and the velocity of the aircraft can be represented as $\vec{V} = V\hat{i}_D$, as seen in Figure B2.

following equations:

$$\mu = \kappa_\mu(\psi_{des} - \psi), \tag{B16}$$

$$C_L = \bar{C}_L + \kappa_{C_L}(\gamma_{des} - \gamma), \tag{B17}$$

$$\dot{T} = \kappa_{T,1}(T_{des} - T), \tag{B18}$$

$$T_{des} = \bar{T} + \kappa_{T,2}(V_{des} - V). \tag{B19}$$

Here, ψ_{des}, γ_{des}, T_{des}, and V_{des} are the desired values of the vehicle's yaw, pitch, thrust, and velocity, respectively. The feed forward terms $\bar{C}_L$ and $\bar{T}$ are the values of C_L and T, respectively, that are required to maintain steady flight:
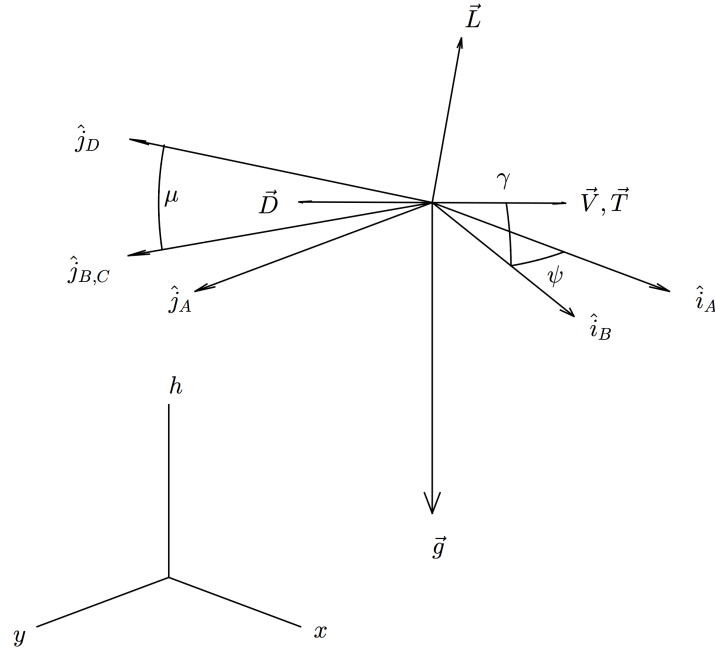
FIGURE B2 Three-dimensional flight: forces and velocity.

$$\bar{C}_L = \frac{2mg\cos(\gamma)}{SV^2\rho}, \tag{B20}$$

$$\bar{T} = mg\sin(\gamma) + \frac{1}{2}C_{D0}SV^2\rho + \frac{2K_d m^2 g^2 \cos(\gamma)^2}{SV^2\rho}. \tag{B21}$$

The vehicle's drag polar parameter is denoted K_d. Similar to the quadrotor model, controller gains κ_μ, κ_{C_L}, $\kappa_{T,1}$, and $\kappa_{T,2}$ may be tuned to better replicate the behavior of a real-world UAS.

The outer-loop controller is separated into two components: longitudinal and lateral. Its design requires us to add two additional states (V_{des} and ψ_{des}) and to estimate $\ddot{\eta}_{des}$ via finite differences, which is undesirable from a computational efficiency perspective, but avoids wrap-around issues.

For the longitudinal controller, V is assumed to be constant and only the $\dot{h}$ dynamics are considered:

$$\dot{h} = V\sin(\gamma), \tag{B22}$$

$$\text{Control}: \gamma, \tag{B23}$$

$$e = h - h_{des}, \tag{B24}$$

$$\dot{e} = -\kappa e, \tag{B25}$$

$$\gamma_{des} = \sin^{-1}\left(\frac{\dot{h}_{des} - \kappa(h - h_{des})}{V}\right). \tag{B26}$$

Note that because above controller is part of the outer-loop, the output is γ_{des} and not γ.

For the lateral controller, we need to control $\dot{\psi}$ so a different approach is pursued. Here, γ is assumed to be constant and only the $\dot{x}$ and $\dot{y}$ kinematics are considered:

$$\dot{x} = V \cos(\gamma) \cos(\psi), \tag{B27}$$

$$\dot{y} = V \cos(\gamma) \sin(\psi), \tag{B28}$$

$$\text{Controls} : \dot{V}, \dot{\psi}, \tag{B29}$$

$$\eta = \begin{bmatrix} x \\ y \end{bmatrix}, \ e = \eta - \eta_{des}, \tag{B30}$$

$$S = \dot{e} + \kappa e, \tag{B31}$$

$$\dot{S} = \ddot{\eta} - \ddot{\eta}_{des} + \kappa \dot{e}, \tag{B32}$$

$$\dot{S} = \begin{bmatrix} \cos(\gamma)\cos(\psi_{des}) & -V_{des}\cos(\gamma)\sin(\psi_{des}) \\ \cos(\gamma)\sin(\psi_{des}) & V_{des}\cos(\gamma)\cos(\psi_{des}) \end{bmatrix} \begin{bmatrix} \dot{V}_{des} \\ \dot{\psi}_{des} \end{bmatrix}$$
$$- \ddot{\eta}_{des} + \kappa \dot{e}, \tag{B33}$$

$$\dot{S} = Av - \ddot{\eta}_{des} + \kappa \dot{e}, \tag{B34}$$

$$\dot{S} = -\Lambda_f S, \ -\Lambda_f \text{ Hurwitz}, \tag{B35}$$

$$\begin{bmatrix} \dot{V} \\ \dot{\psi} \end{bmatrix} = A^{-1}[\ddot{\eta}_{des} - \kappa \dot{e} - \Lambda_f S]. \tag{B36}$$

This control is invalid if A is singular, corresponding to a pitch angle of $\pm 90°$ or a desired velocity of $V_{des} = 0$, which are flight conditions not encountered in this work.

For the wind disturbance, the Dryden wind model defines wind in the longitudinal, lateral, and vertical body fixed directions, *not* in the x, y, h directions. In the two dimensional case[40], we assumed that γ is small and thus we have wind in the x and h directions. In the three-dimensional case, however, we cannot necessarily make the same assumptions about ψ. Thus, the following wind definitions are used:

$$w = w_u \hat{i}_D + w_w \hat{j}_D + w_v \hat{k}_D = w_x \hat{i}_A + w_y \hat{j}_A + w_h \hat{k}_A. \tag{B37}$$

We can now write the full equations of motion as follows:

$$\dot{x} = V \cos(\gamma) \cos(\psi) + w_x, \tag{B38}$$

$$\dot{y} = V \cos(\gamma) \sin(\psi) + w_y, \tag{B39}$$

$$\dot{h} = V \sin(\gamma) + w_h, \tag{B40}$$

$$\dot{V} = \frac{T - D}{m} - g \sin(\gamma) - \dot{w}_x \cos(\gamma) \cos(\psi)$$
$$- \dot{w}_y \cos(\gamma) \sin(\psi) + \dot{w}_h \sin(\gamma), \tag{B41}$$

$$\dot{\psi} = \frac{-1}{V m \cos(\gamma)} \Big[L \sin(\mu) - m \dot{w}_x \sin(\psi)$$
$$+ m \dot{w}_y \cos(\psi) \Big], \tag{B42}$$

$$\dot{\gamma} = \frac{1}{V m} \Big[L \cos(\mu) - mg \cos(\gamma) + m \dot{w}_x \cos(\psi) \sin(\gamma)$$
$$+ m \dot{w}_y \sin(\gamma) \sin(\psi) + m \dot{w}_h \cos(\gamma) \Big], \tag{B43}$$

where

$$w_x = w_u \cos(\gamma) \cos(\psi) - w_w(\cos(\mu) \sin(\psi)$$
$$+ \cos(\psi) \sin(\gamma) \sin(\mu)) - w_v(\sin(\mu) \sin(\psi)$$
$$- \cos(\mu) \cos(\psi) \sin(\gamma)), \tag{B44}$$

$$w_y = w_v(\cos(\psi) \sin(\mu) + \cos(\mu) \sin(\gamma) \sin(\psi))$$
$$+ w_w(\cos(\mu) \cos(\psi) - \sin(\gamma) \sin(\mu) \sin(\psi))$$
$$+ w_u \cos(\gamma) \sin(\psi), \tag{B45}$$

$$w_h = w_v \cos(\gamma) \cos(\mu) - w_u \sin(\gamma)$$
$$- w_w \cos(\gamma) \sin(\mu), \tag{B46}$$

$$C_D = C_{D0} + K_d C_L^2, \tag{B47}$$

$$L = \frac{1}{2} C_L \rho S V^2, \tag{B48}$$

$$D = \frac{1}{2} C_D \rho S V^2. \tag{B49}$$

If we assume that $|\dot{w}_u|, |\dot{w}_w|, |\dot{w}_v| \gg |\dot{\gamma}|, |\dot{\psi}|, |\dot{\mu}|$ then we can express the wind accelerations as follows:

$$\dot{w}_x = \dot{w}_u \cos(\gamma) \cos(\psi) - \dot{w}_w(\cos(\mu) \sin(\psi)$$
$$+ \cos(\psi) \sin(\gamma) \sin(\mu)) - \dot{w}_v(\sin(\mu) \sin(\psi)$$
$$- \cos(\mu) \cos(\psi) \sin(\gamma)), \tag{B50}$$

$$\dot{w}_y = \dot{w}_v(\cos(\psi) \sin(\mu) + \cos(\mu) \sin(\gamma) \sin(\psi))$$
$$+ \dot{w}_w(\cos(\mu) \cos(\psi) - \sin(\gamma) \sin(\mu) \sin(\psi))$$
$$+ \dot{w}_u \cos(\gamma) \sin(\psi), \tag{B51}$$

$$\dot{w}_h = \dot{w}_v \cos(\gamma) \cos(\mu) - \dot{w}_u \sin(\gamma)$$
$$- \dot{w}_w \cos(\gamma) \sin(\mu). \tag{B52}$$

The power spectral densities (PSD) for the Dryden model are defined following[8].
For the longitudinal channel:

$$\Phi_u(\Omega) = \sigma_u^2 \frac{2L_u}{\pi} \frac{1}{[1 + L_u^2 + \Omega^2]^2}, \tag{B53}$$

For the lateral channel:

$$\Phi_w(\Omega) = \sigma_w^2 \frac{L_w}{\pi} \frac{1 + 3L_w^2 \Omega^2}{[1 + L_w^2 \Omega^2]^2}, \tag{B54}$$

For the vertical channel:

$$\Phi_v(\Omega) = \sigma_v^2 \frac{L_v}{\pi} \frac{1 + 3L_v^2 \Omega^2}{[1 + L_v^2 \Omega^2]^2}, \tag{B55}$$

where $\Omega = \frac{\omega}{V}$. The corresponding coloring filters are as follows:

$$G_u(s) = \sigma_u \sqrt{\frac{2L_u}{\pi V}} \frac{1}{1 + \frac{L_u}{V}s}, \tag{B56}$$

$$G_w(s) = \sigma_w \sqrt{\frac{L_w}{\pi V}} \frac{1 + \frac{\sqrt{3}L_w}{V}s}{\left(1 + \frac{L_w}{V}s\right)^2}, \tag{B57}$$

$$G_v(s) = \sigma_v \sqrt{\frac{L_v}{\pi V}} \frac{1 + \frac{\sqrt{3}L_v}{V}s}{\left(1 + \frac{L_v}{V}s\right)^2}. \tag{B58}$$

These have state space realizations of the form:

$$\dot{\eta}_i = A_i \eta_i + B_i n, \tag{B59}$$

$$w_i = C_i \eta_i, \tag{B60}$$

$$\dot{w}_i = C_i A_i \eta_i + C_i B_i n, \tag{B61}$$

for $i \in \{u, w, v\}$, where n is the Gaussian white noise input and the matrices are defined as follows:

$$A_u = \frac{-V}{L_u}, A_w = \begin{bmatrix} \frac{-2V}{L_w} & \frac{-V^2}{L_w^2} \\ 1 & 0 \end{bmatrix}, A_v = \begin{bmatrix} \frac{-2V}{L_v} & \frac{-V^2}{L_v^2} \\ 1 & 0 \end{bmatrix}, \tag{B62}$$

$$B_u = 1, B_w = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, B_v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{B63}$$

$$C_u = \frac{\sqrt{2}V\sigma_u\sqrt{\frac{L_u}{V}}}{L_u\sqrt{\pi}}, C_w = \frac{V\sigma_w\sqrt{\frac{L_w}{V}}}{L_w\sqrt{\pi}} \begin{bmatrix} \sqrt{3} & \frac{V}{L_w} \end{bmatrix},$$

$$C_v = \frac{V\sigma_v\sqrt{\frac{L_v}{V}}}{L_v\sqrt{\pi}} \begin{bmatrix} \sqrt{3} & \frac{V}{L_v} \end{bmatrix}. \tag{B64}$$

References

1. Schaufele Jr RD, Corning J, Ding L, et al. FAA Aerospace Forecast: Fiscal Years 2018-2038. *Washington, DC* 2018.

2. Kopardekar P, Rios J, Prevot T, Johnson M, Jung J, Robinson J. Unmanned aircraft system traffic management (UTM) concept of operations. tech. rep., NASA; : 2016.

3. Dill ET, Young SD, Hayhurst KJ. SAFEGUARD: An assured safety net technology for UAS. In: 2016 IEEE/AIAA 35th digital avionics systems conference (DASC). IEEE. ; 2016: 1–10.

4. D'Souza S. Developing a generalized trajectory modeling framework for small UAS performance in the presence of wind. In: AIAA Information Systems-AIAA Infotech @ Aerospace, AIAA SciTech Forum. ; 2017.

5. Richardson JR, Atkins EM, Kabamba PT, Girard AR. Envelopes for flight through stochastic gusts. *Journal of Guidance, Control, and Dynamics* 2013; 36(5): 1464–1476.

6. Belcastro CM, Foster JV. Aircraft loss-of-control accident analysis. In: Proceedings of AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Paper No. AIAA-2010-8004. ; 2010.

7. Administration FA. *Review of Aviation Accidents Involving Weather Turbulence in the United States 1992-2001*. Aviation Safety Data Analysis Center . 2004.

8. Hoblit FM. Gust Loads on Aircraft: Concepts and Applications, AIAA Education Series, American Institute of Aeronautics and Astronautics. *Inc., Washington, DC* 1988.

9. Richardson JR. *Quantifying and scaling airplane performance in turbulence*. PhD thesis. University of Michigan, Ann Arbor; 2013.

10. Liepmann HW. On the application of statistical concepts to the buffeting problem. *Journal of the Aeronautical Sciences* 1952.

11. De Karman T, Howarth L. On the statistical theory of isotropic turbulence. In: . 164.917 of *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. The Royal Society. ; 1938: 192–215.

12. Von Karman T. Progress in the statistical theory of turbulence. *Proceedings of the National Academy of Sciences* 1948; 34(11): 530–539.

13. Von Kármán T, Lin C. On the statistical theory of isotropic turbulence. *Advances in Applied Mechanics* 1951; 2: 1–19.

14. Diederich F, Drischler J. Effect of Spanwise Variations in Gust Intensity on the Lift due to Atmospheric Turbulence. National Advisory Committee for Aeronautics. Tech. Rep. Technical Note 3920, NASA; : 1957.

15. Junkins JL, Akella MR, Alfrined KT. Non-Gaussian error propagation in orbital mechanics. *Guidance and Control* 1996: 283–298.

16. Sabol C, Hill K, Alfriend K, Sukut T. Nonlinear effects in the correlation of tracks and covariance propagation. *Acta Astronautica* 2013; 84: 69–80.

17. Geller DK. Linear covariance techniques for orbital rendezvous analysis and autonomous onboard mission planning. *Journal of Guidance Control and Dynamics* 2006; 29(6): 1404–1414.

18. Geller DK, Rose MB, Woffinden DC. Event triggers in linear covariance analysis with applications to orbital rendezvous. *Journal of Guidance, Control, and Dynamics* 2009; 32(1): 102.

19. Julier SJ, Uhlmann JK, Durrant-Whyte HF. A new approach for filtering nonlinear systems. In: . 3 of *American Control Conference, Proceedings of the 1995*. IEEE. ; 1995: 1628–1632.

20. Julier S, Uhlmann J, Durrant-Whyte HF. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on automatic control* 2000; 45(3): 477–482.

21. Wiener N. The homogeneous chaos. *American Journal of Mathematics* 1938; 60(4): 897–936.

22. Terejanu G, Singla P, Singh T, Scott PD. Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models. *Journal of Guidance, Control, and Dynamics* 2008; 31(6): 1623.

23. DeMars KJ, Bishop RH, Jah MK. Entropy-based approach for uncertainty propagation of nonlinear dynamical systems. *Journal of Guidance, Control, and Dynamics* 2013.

24. Vittaldev V, Russell RP. Space object collision probability using multidirectional gaussian mixture models. *Journal of Guidance, Control, and Dynamics* 2016.

25. Luo Yz, Yang Z. A review of uncertainty propagation in orbital mechanics. *Progress in Aerospace Sciences* 2017; 89: 23–39.

26. Blackmore L, Ono M, Williams BC. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics* 2011; 27(6): 1080–1094.

27. Silva Arantes dM, Toledo CFM, Williams BC, Ono M. Collision-Free Encoding for Chance-Constrained Nonconvex Path Planning. *IEEE Transactions on Robotics* 2019; 35(2): 433–448.

28. Bry A, Roy N. Rapidly-exploring random belief trees for motion planning under uncertainty. In: 2011 IEEE international conference on robotics and automation. IEEE. ; 2011: 723–730.

29. Janson L, Schmerling E, Pavone M. Monte Carlo motion planning for robot trajectory optimization under uncertainty. In: Robotics Research. Springer. 2018 (pp. 343–361).

Author Manuscript

30. Ding XC, Pinto A, Surana A. Strategic planning under uncertainties via constrained Markov decision processes. In: 2013 IEEE International Conference on Robotics and Automation. IEEE. ; 2013: 4568–4575.

31. Luders B, Kothari M, How J. Chance constrained RRT for probabilistic robustness to environmental uncertainty. In: AIAA Guidance, Navigation, and Control Conference. AIAA. ; 2010: 8160.

32. Pairet È, Hernández JD, Lahijanian M, Carreras M. Uncertainty-based online mapping and motion planning for marine robotics guidance. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. ; 2018: 2367–2374.

33. Axelrod B, Kaelbling LP, Lozano-Pérez T. Provably safe robot navigation with obstacle uncertainty. *The International Journal of Robotics Research* 2018; 37(13-14): 1760–1774.

34. Kothari M, Postlethwaite I. A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees. *Journal of Intelligent & Robotic Systems* 2013; 71(2): 231–253.

35. Borowski H, Frew E. An evaluation of path planners for guidance with vision based simultaneous localization and mapping. In: AIAA Guidance, Navigation, and Control Conference. AIAA. ; 2012: 4906.

36. LaValle SM. Rapidly-exploring random trees: A new tool for path planning. Online; 1998. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.1853&rep=rep1&type=pdf, Accessed November 21, 2019.

37. Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 2011; 30(7): 846–894.

38. Gammell JD, Srinivasa SS, Barfoot TD. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE. ; 2014: 2997–3004.

39. Adiyatov O, Varol HA. A novel RRT*-based algorithm for motion planning in Dynamic environments. In: Mechatronics and Automation (ICMA), 2017 IEEE International Conference on. IEEE. ; 2017: 1416–1421.

40. Berning AW, Taheri E, Girard A, Kolmanovsky I. Rapid uncertainty propagation and chance-constrained trajectory optimization for small unmanned aerial vehicles. In: 2018 Annual American Control Conference (ACC). IEEE. ; 2018: 3183–3188.

41. Kabamba PT, Girard AR. *Fundamentals of Aerospace Navigation and Guidance*. Cambridge University Press . 2014.

42. Malyshev VV, Krasilshikov MN, Karlov VI. *Optimization of observation and control processes*. AIAA . 1992.

43. Lancaster HO, Seneta E. *Chi-square distribution*. Wiley Online Library . 1969.

44. Hanson FB. *Applied stochastic processes and control for jump-diffusions: modeling, analysis, and computation*. 13. Siam . 2007.

45. Waslander S, Wang C. Wind disturbance estimation and rejection for quadrotor position control. In: AIAA Infotech@Aerospace Conference and AIAA Unmanned Unlimited Conference. AIAA. ; 2009: 1983.

46. Hull DG. *Fundamentals of airplane flight mechanics*. Springer . 2007.