NASA/TP–20240013241

# Quantification of Uncertainty and Risk Sensitivity for Safety of Emerging Operations

*Rajnish Bhusal, Aakarshan Khanal, Kamesh Subbarao, Animesh Chakravarthy*
*The University of Texas at Arlington, Arlington, Texas*

*Wendy A. Okolo*
*NASA Ames Research Center*

October 2024

NASA/TP–20240013241

# Quantification of Uncertainty and Risk Sensitivity for Safety of Emerging Operations

*Rajnish Bhusal, Aakarshan Khanal, Kamesh Subbarao, Animesh Chakravarthy*
*The University of Texas at Arlington, Arlington, Texas*

*Wendy A. Okolo*
*NASA Ames Research Center*

National Aeronautics and
Space Administration

AMES Research Center
Mountain View, California 94035

October 2024

## Acknowledgments

This work was supported by the System-Wide Safety Project under NASA's Airspace Operations and Safety Program. The authors would like to thank Dr. Mateo Corbetta for feedback provided during the course of the project that helped us refine the technical material particularly considering data based uncertainty characterization. The authors would also like to thank Alexandra E. Jannetta for the administrative support received during the course of the project.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an offical endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://www.sti.nasa.gov

# Quantification of Uncertainty and Risk Sensitivity for Safety of Emerging Operations

## Summary

The growing need to develop and deploy small unmanned aerial vehicles (sUAVs) for various applications in the airspace necessitates reliable tools to accurately predict the flight trajectories of the sUAVs. The knowledge of the predicted trajectories help decision makers anticipate potential conflict, assess the risk, and take appropriate risk mitigation actions. In addition, uncertainties in vehicle models, weather, and controller action further highlights the need for reliable prediction tools.

In this project, the application of mixed sparse grid-based quadrature and generalized polynomial chaos (gPC) expansion method for uncertainty quantification and collision assessment in air traffic consisting of fixed-wing small unmanned aerial vehicles (sUAV) was studied. From the results obtained, it can be concluded that this provides a reliable framework to carry out quantitative conflict assessment in an unmanned air traffic, which when employed, can improve the functionalities of the unmanned traffic management system. It was observed that the results from the gPC expansion framework developed in the project can be utilized to conduct rapid probabilistic collision assessment for near real-time unmanned traffic management in the airspace. From the vehicle models, position updates, and wind-field data, apriori gPC based 3-$\sigma$ confidence ellipses can provide estimates of potential conflict at some future instants. The computational costs scaled linearly when the uncertain inputs were fewer. Further, the largest allowable distribution of parametric uncertainties that leads to the smallest risk of collision in traffic of small unmanned aerial vehicles could be calculated. The time of closest approach between two sUAVs can be established paving way for development of proactive mitigation strategies. The separation between the sUAVs was found to be most significantly affected by uncertainties in the maximum available thrusts, zero-lift drag coefficients, and wing planform areas of the sUAVs. The study of uncertain wind-fields indicated that a heterogeneous traffic mix resulted in an increased probability of conflict. Increased measurement update rate reduced the uncertainties in the trajectories of the vehicles, further reducing the probability of conflict but rapid updates of all vehicles in the airspace poses a stringent communication limitation. The gPC framework also provided the means to analyze vehicle impact (crash region) due to loss of control resulting from actuator failure in sUAS traffic, essentially to predict impact and crash zones for representative vehicles. The predicted regions when compared with non-participant density, provides a means to develop an early mitigation strategy, should the sUAV detect an imminent actuator failure.

# Contents

# 1 Introduction

In recent years, small unmanned aerial vehicles (sUAVs) have been used for civilian, commercial, and military applications including real-time monitoring of infrastructures, remote sensing, search and rescue operations, delivery of goods, precision agriculture, and photography. This widespread use of sUAVs in low-altitude crowded airspace will create new challenges in terms of conflict risk which demands robust unmanned traffic management (UTM) functionalities. To that end, NASA's Aeronautics Research Mission Directorate is actively investigating and advancing in-time system-wide safety assurance (ISSA) capabilities for unmanned traffic management Ref. 1. It is envisioned that advanced safety assurance tools will leverage vast amounts of aviation system data to allow for the timely identification of precursors, anomalies, and trends thereby maintaining acceptable safety margins Ref. 2. Thus, there is a strong demand for tools and techniques that provide timely detection, diagnosis, and predictive capabilities regarding the risks and hazards associated with the in-flight operation of these sUAVs. A recent survey of UTM and the tools and techniques available for conflict risk assessment in traffic of unmanned aerial vehicles can be found in Ref. 3.

To conduct reliable UTM functionalities and specifically to analyze and predict the safety of the overall airspace, it is essential to know the current and future locations of all sUAVs in the airspace. As a result, trajectory prediction for unmanned aerial systems has been an active research topic over the past decades Ref. 4–6. In general, for UTM, the trajectories of the sUAVs are planned beforehand, also known as filed flight plan. However, the systems of sUAVs are not free from uncertainties in the vehicle parameters and atmospheric conditions. Due to these uncertainties, the actual trajectories of the vehicles may deviate significantly from the filed flight plan, which may increase the risk of conflict in the airspace. It can be argued that a better understanding and quantification of such uncertainties enable better predictions of the vehicle trajectories in the airspace Ref. 7. Therefore, it is essential to analyze and quantify the various sources of uncertainty that affect the airspace, systematically quantify their impact on the UTM operations, estimate the effect of uncertainty on safety, and aid risk-informed decision-making activities to ensure smooth operation of the overall airspace Ref. 8. Further, conducting the simulation and prognostics in a probabilistic manner can help risk mitigation and decision-making in UTM. For recent surveys on conflict detection and resolution methods for the traffic of manned and unmanned aerial vehicles, please refer to Ref. 9

In the literature, the most common approach to carry out uncertainty quantification and risk analysis for sUAVs is through Monte Carlo-based repetitive simulations. This involves exhaustive random sampling of uncertain variables within the domain of their probability density functions (PDFs). For example, Latin Hypercube sampling-based Monte Carlo technique and sequential Monte Carlo technique are employed in Ref. 6 and Ref. 10, respectively to carry out uncertainty quantification and propagation in predicting trajectories of sUAVs in the airspace. The Monte Carlo methods, while being highly accurate to estimate statistical moments as well as PDFs of the quantities of interest, are computationally expensive for systems with large dimensions of uncertainties and may become intractable for computationally demanding models Ref. 11. The high computational cost of traditional Monte Carlo-based methods has motivated the development of computationally efficient meta-modeling approaches for uncertainty quantification and propagation Ref. 12, 13.

In recent years, the generalized polynomial chaos (gPC) expansion framework has been extensively used as a meta-modeling technique to carry out uncertainty quantification and propagation Ref. 14–16. The gPC expansion is based on the principle of projecting the random solution onto a basis of polynomials which are orthogonal to the PDF of the input random variables. The orthogonal polynomials for the random variables are selected from the Askey scheme (see Ref. 17) based on the distribution of the random variables.

Ref. 16 shows the use of the gPC expansion technique for uncertainty propagation and trajectory prediction in unmanned aerial vehicles. Among other applications of this method, gPC expansion was utilized to propagate uncertainty and to estimate the probability of collision of spacecraft in Ref. 18, and in uncertainty propagation for cooperative teams of unmanned vehicles (see Ref. 19).

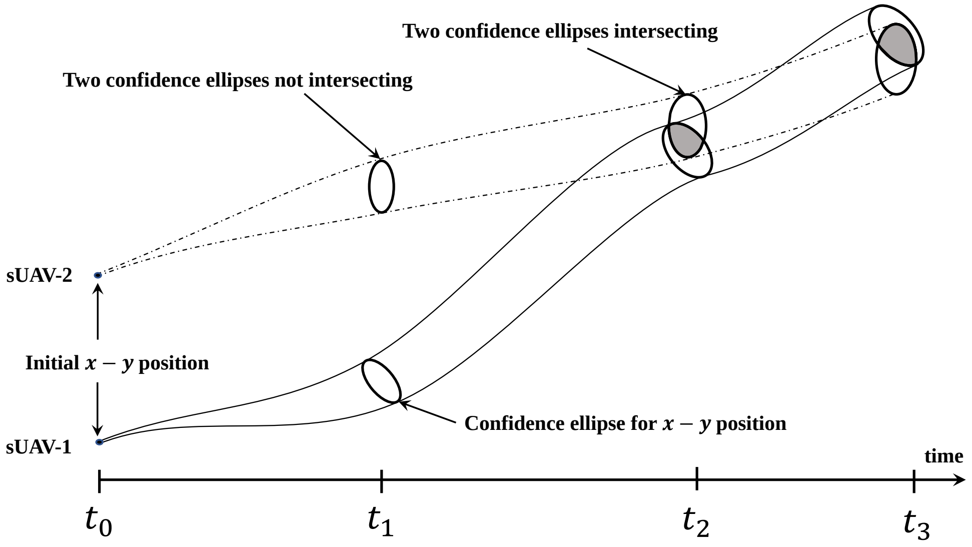

Two confidence ellipses intersecting
Two confidence ellipses not intersecting
sUAV-2
Initial $x-y$ position
sUAV-1
Confidence ellipse for $x-y$ position
time
$t_0$
$t_1$
$t_2$
$t_3$

**Figure 1.—**Illustration of confidence ellipses of position for two sUAVs demonstrating the risk of conflict

Figure 1 depicts an illustrative sketch of the temporal variation in the position uncertainty for two sUAVs. In Fig. 1, while both the sUAVs have deterministic initial conditions, the models of the sUAVs are subject to parametric uncertainties. This results in uncertainties in the trajectories of sUAVs with time. To estimate the uncertainties in the position, one can rely upon various uncertainty quantification and propagation algorithms available in the literature. Further, one can compute the statistics of the position trajectories in terms of mean and co-variance and compute the confidence region of the trajectories. These confidence regions provide an estimate of the location of the sUAV at a particular instant with some corresponding probability. For instance, $3-\sigma$ confidence intervals (in one dimension) and $3-\sigma$ confidence ellipse (in two dimensions) provide an estimate of the sUAV location with 99.7 % probability of occurrence (see Ref. 20). In Fig. 1, the uncertainties in the $X-Y$ trajectories of the sUAVs at constant altitude are represented by $3-\sigma$ confidence regions of elliptical cross-sections. Also note the shaded regions in Fig. 1 at time $t_2$ and $t_3$, which depict the intersection in the $3-\sigma$ confidence ellipses of the two sUAVs at those time instants implies that there is a risk of conflict between them.

In this publication, we first consider the traffic of sUAVs in airspace with probabilistic uncertainties in their initial positions, speeds, and model parameters. Subsequently, we extend our study to include wind-field uncertainties in the airspace. The uncertainty propagation in the trajectories of the sUAVs is carried out using the gPC expansion technique. We assume that the uncertainties in the initial conditions and parameters have different probability density types (normally distributed initial condition uncertainties and uniformly distributed parameters). To address the mixed uncertainty types, we use the framework of pseudospectral collocation with a mixed sparse grid quadrature technique in Ref. 13 to compute the PDF contours and statistics (mean and $3-\sigma$ confidence regions) of the trajectories of the sUAVs. The confidence regions are then helpful in approximating the probability of conflict between any two vehicles in the traffic of sUAVs.

Since, the gPC expansion-based uncertainty quantification framework provides the response distribution and statistics of the trajectories of sUAVs we also seek to answer the following question, "How much does each of the individual uncertainties contribute to the overall uncertainty in the trajectories of sUAVs?" (see Ref. 6 and Ref. 19). To that end, we look at the polynomial chaos-based Sobol' indices, proposed in Ref. 21, for sensitivity analysis. Specifically, we study the minimum separation distance between sUAVs and quantify the sensitivity of this separation distance with respect to the underlying uncertainties in the initial conditions, and parameters of the vehicles involved.

In the present context, there has been an emerging use of different models (heterogeneous) of sUAVs in low-altitude congested airspace (see Ref. 22). In addition to the uncertainties mentioned above, we also study the air traffic subject to uncertain wind-fields (see Ref. 23). Using Monte Carlo simulations, Ref. 22, studied the UAVs' conflict detection systems in densely populated airspace. Similarly, Ref. 6 used Latin Hypercube sampling-based Monte Carlo simulation to carry out uncertainty quantification in predicting the trajectory of an aircraft subject to wind uncertainties. In Ref. 24, the authors develop a statistical model of wind forecast uncertainty and apply it to stochastic predictions of level flight trajectories of an aircraft using Monte Carlo-based technique. Further, Monte Carlo simulations were also used in Ref. 25 and Ref. 26 to examine the effect of wind uncertainties on conflict probabilities in traffic of aircraft. However, it is well known that Monte Carlo simulation-based techniques have a very slow mean convergence rate and demand a large number of sample points to obtain accurate results, thereby being computationally cumbersome for uncertainty propagation in traffic of vehicles (Ref. 27). In the context of wind uncertainty effects on UAVs, Ref. 16 compares the computational efficacy of gPC expansion-based aircraft trajectory prediction with that of the Monte Carlo approach. Moreover, Ref. 27 uses the gPC expansion technique to develop a conflict detection algorithm among multiple heterogeneous aircraft.

Different from most of the works in the literature including the aforementioned, where the entire wind field (or wind velocity components) is modeled as a stochastic process, we consider the uncertainties in the spatial gradients of the wind field. The approach of pseudospectral collocation-based gPC expansion in Ref. 13 and Ref. 12 is utilized for uncertainty propagation in the heterogeneous models of the UAVs, comprising fixed-wing UAVs and quadrotors in confined airspace. Within this context, we quantify the risk of conflict in the airspace due to wind field uncertainties, by calculating the probability that the separation distance between any two UAVs would be smaller than the tolerable safety limit at any future instant of time.

It should be noted that the $3-\sigma$ confidence ellipses of the vehicles' trajectories at any instant is dependent on the parametric uncertainties and proportional to the size of these uncertainties. Intuitively, the risk of conflict between any two sUAVs (or alternatively, the intersection of $3-\sigma$ confidence ellipses of two sUAVs at any instant) increases with an increase in the size of the uncertainties in the underlying parameters. In the probabilistic sense, the size of the uncertainty associated with a parameter is governed by its probability distribution. Therefore, it is essential to estimate the maximum distribution of uncertainties in the parameters of the sUAVs for their safe operation in the airspace, which is the key motivation behind the work carried out in this publication.

To that end, this publication considers traffic of stochastic small unmanned aerial vehicles flying at constant altitude such that the trajectory of the vehicles can be represented by two-dimensional position coordinates. Next, we develop a computational tool to compute the maximum distribution of parametric uncertainties such that the $3-\sigma$ confidence ellipses of the sUAVs' trajectories do not intersect with each other throughout the flight time. We restrict the PDFs of the uncertainties to being Gaussian and/or Uniform in the present study. In the context of this publication, we define the maximum distribution as follows: (i) maximum standard deviation for normally distributed parameters, and (ii) maximum difference between upper and lower bounds for uniformly distributed parameters. This problem is solved as a constrained multi-variable multi-objective optimization. Further, we impose ellipsoidal intersection conditions presented in

Ref. 28 and use them as mathematical constraints in the optimization.

While most hitherto discussion hinged on UQ using functional representations of the uncertainties, it is important to recognize that there is a need to quantify uncertainty from actual data (measurements). To address this challenge, we investigate the use of a machine learning technique (Gaussian Process Regression) and show its utility for UQ. Finally the publication presents an open-source Python toolbox for gPC expansion of dynamic systems with initial conditions and parametric uncertainties. The toolbox supports analysis of Gaussian and Uniformly distributed uncertainties and can be used to perform UQ when there is a mix of such uncertainties present in the dynamical systems.

The publication is organized as follows. Section 2 presents the Uncertainty Quantification framework using the generalized Polynomial Chaos expansion technique briefly. Section 3 describes the vehicle models and the controllers utilized to generate synthetic data and verify the UQ framework. Section 4 applies the UQ framework for conflict assessment of air traffic with fixed-wing UAVs in steady conditions with no wind-fields. Section 5 discusses the application of the conflict assessment in the presence of wind-field uncertainties. Section 6 provides a methodology to estimate the probability of conflict (confidence region) leading to Section 7 which presents a method to determine the maximum uncertainty bounds on key vehicle parameters to ensure a conflict free traffic of small UAVs. Section 8 extends the UQ analysis to evaluate crash zones resulting from loss of control of the UAVs. In section 9, we adopt a Machine Learning based approach to characterize uncertainties from synthetic data to address scenarios where probability distributions of uncertain input variables is unavailable or impractical. Section 10 presents conclusions from this research.

# 2 Uncertainty Quantification using Generalized Polynomial Chaos Expansion

## 2.1 Preliminaries and Problem Statement

In this section, we provide a theoretical background on orthogonal polynomials and stochastic processes and formulate the problem under consideration.

### 2.1.1 Polynomial approximations and Orthogonal Polynomials

In approximation theory, it is common to accurately approximate an arbitrary function with a finite sum of polynomials. In that context, orthogonal polynomials are widely employed for polynomial approximations. Consider a set of polynomials, $\{Q_n(x), n \in \mathbb{N}\}$ where $\mathbb{N} = 0, 1, 2, \ldots$ and $Q_n(x)$ is a polynomial in $x$ of degree $n$ of the following form

$$Q_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0, \quad a_n \neq 0. \tag{1}$$

The system of polynomials $\{Q_n(x), n \in \mathbb{N}\}$ is called an orthogonal system of polynomials with respect to a weight function $w(x)$ on $(a, b)$ if it satisfies following orthogonality relation

$$\int_a^b Q_n(x) Q_m(x) w(x) dx = \gamma_n \delta_{mn}, \quad m, n \in \mathbb{N} \tag{2}$$

where $\delta_{mn}$ is the Kronecker delta function ($\delta_{mn} = 0$ if $m \neq n$ and $\delta_{mn} = 1$ if $m = n$) and $\gamma_n > 0$ are the normalization constants given by

$$\gamma_n = \int_a^b Q_n^2(x) w(x) dx. \tag{3}$$

One of the most important characteristics of a classical orthogonal system of polynomials $\{Q_n(x)\}$ is that any three consecutive polynomials in the system satisfy the following recurrence relation, well known as three-term recurrence relation Ref. 29

$$Q_{n+1}(x) = (A_n x + B_n)Q_n(x) - C_n Q_{n-1}(x), \quad n \geq 0 \tag{4}$$

where $A_n$, $B_n$, and $C_n$ are arbitrary sequences of real numbers such that $A_n \neq 0$, $C_n \neq 0$, and $C_n A_n A_{n-1} > 0$ for all $n \in \mathbb{N}$.

### 2.1.2 Probability Space and Random Vector

Let $(\mathit{\Omega}, \mathcal{F}, \mathcal{P})$ be a probability space, where $\mathit{\Omega}$ is the sample space, $\mathcal{F}$ is the $\sigma$-algebra of the subsets of $\mathit{\Omega}$, and $\mathcal{P}$ is the probability measure. Please refer to Ref. 30 for details on the probability space. For a random event $\omega \in \mathit{\Omega}$, let $Z = Z(\omega)$ be a continuous random variable which is generally described by a probability density function (PDF). This report focuses on random variables with two types of probability distributions, Gaussian and uniform. For the Gaussian (also known as normal) distribution of random variable $Z$, we denote the PDF as $Z \sim \mathcal{N}(\mu, \sigma^2)$ where $\mu$ is the mean and $\sigma$ is the standard deviation. On the other hand, the uniformly distributed random variable $Z$ has a PDF of form $Z \sim \mathcal{U}[a, b]$ with $a$ and $b$ as lower and upper bounds of the uniform PDF, respectively. The cumulative density function (CDF) of a random variable $Z$ can be defined as $F_Z(z) = \mathcal{P}(Z < z)$.

Further, for the $d$-dimensional probability space with random variables $Z_1, Z_2, \ldots, Z_d$, $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_d]^{\mathrm{T}} \in \mathbb{R}^d$ denotes a random vector. In this work, we describe any system as a stochastic system if the parameters of the system are uncertain and parameterized by the random vector $\mathbf{Z}$.

### 2.1.3 Problem Statement

Let us consider a traffic of small unmanned aerial vehicles (sUAVs) such that the dynamics of each vehicle are modeled by a stochastic differential equation (SDE) of the form

$$\dot{\mathbf{x}}(t, \mathbf{Z}) = \mathbf{f}(t, \mathbf{x}, \mathbf{p}, \mathbf{Z}), \quad \mathbf{x}_0 = \mathbf{x}(0, \mathbf{Z}_{x_0}) \tag{5}$$

where $\mathbf{x} = [x_1, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the state vector, $t \in [0, T]$ is the temporal variable, $\mathbf{p} \in \mathbb{R}^p$ is the vector of model parameters, and $\mathbf{x}_0$ is the initial state vector at time $t = 0$. Here $\mathbf{Z} \in \mathbb{R}^d$, $d \leq n + n_p$ denotes the random vector which represents the uncertainties in the system in Eq. (5).

In this section, we consider the uncertainties in the initial conditions and parameters of the system. We consider the random vector $\mathbf{Z}$ in the form, $\mathbf{Z} = [\mathbf{Z}_{x_0}^{\mathrm{T}}, \mathbf{Z}_p^{\mathrm{T}}]^{\mathrm{T}}$ where $\mathbf{Z}_{\mathbf{x}_0}$ and $\mathbf{Z}_{\mathbf{p}}$ represent a vector of uncertain initial conditions of the state and model parameters of the system with known stationary probability density functions (PDF), respectively. The model parameters in this case relate to those in the governing differential equations for the vehicle dynamics. Estimates of the uncertainties in terms of their expected values and the variances are typically available from experiments and apriori characterization. For example, it is reasonable to expect the mass of a vehicle to be a uniformly distributed variable. These values are then used to construct the probability density functions. The stationary PDF refers to constant variance in our work. The problems modeled here correspond to those when the random variables pertaining to the uncertain initial conditions and parameters have different distribution types. Specifically, this report carries out uncertainty quantification for stochastic dynamical systems whose random inputs are modeled such that, $\mathbf{Z}_{\mathbf{x}_{0_i}} \sim \mathcal{N}(\mu_i, \sigma_i^2)$ for $i = 1, \ldots, n$ and $\mathbf{Z}_{\mathbf{p}_j} \sim \mathcal{U}[a_j, b_j]$ for $i = 1, \ldots, n_p$. Since Gaussian functions have infinite tails, it is unreasonable to expect the mass or the area of a vehicle to be a Gaussian distributed variable. As such, we assume the vehicle parameters to be Uniformly distributed. On the other hand, it is reasonable to expect the initial conditions (position and velocity), to be a Gaussian random variables with a compact support. While the framework is general enough to accommodate mixed distribution types, we make the specific identification that the state initial conditions are Gaussian random variables while the model parameters are Uniform random variables.

The objective of this section is to quantify the uncertainties in the mathematical model of sUAVs and study the propagation of uncertainties with time. To that end, we wish to compute the complete multi-variate PDF and the statistics (mean and $3-\sigma$ confidence region) of the trajectories of the sUAVs and carry out probabilistic conflict assessment using the approach of generalized polynomial chaos expansion. In addition, we wish to carry out a sensitivity analysis of functions of interest to the underlying uncertain variables.

## 2.2 Generalized Polynomial Chaos Expansion

Generalized polynomial chaos (gPC) expansion provides a framework to propagate uncertainties in the SDE of the form as in Eq. (5). The idea behind gPC expansion is to carry out spectral expansions of the functions with random variables via orthogonal polynomials. In this section, we provide a background of the gPC expansion technique.

For the system in Eq. (5), assuming $\mathbf{x}(t,\mathbf{Z}) = [x_1(t,\mathbf{Z}),\ldots,x_n(t,\mathbf{Z})]^{\mathrm{T}}$ to be a second-order process, the gPC expansion of $x_i(t,\mathbf{Z})$ for each $i = 1,\ldots,n$ can be written as,

$$x_i\left(t,\mathbf{Z}\right) = \sum_{|\mathbf{r}|=0}^{\infty} x_{i,\mathbf{r}}^{c}(t)\mathit{\Phi}_{\mathbf{r}}\left(\mathbf{Z}\right), \qquad \mathit{\Phi}_{\mathbf{r}}\left(\mathbf{Z}\right) = \prod_{j=1}^{d}\mathit{\Phi}_{r_j}(Z_j) \tag{6}$$

where $x_{i,\mathbf{r}}^{c}(t)$ is the coefficient of the multidimensional basis $\mathit{\Phi}_{\mathbf{r}}\left(\mathbf{Z}\right)$, $\mathit{\Phi}_{r_j}(Z_j)$, $j = 1,\ldots,d$ is the $j$th univariate basis in random variable $Z_j$ , and $\mathbf{r} = (r_1,\ldots,r_d)$ is the ordered set of multi-indices with $|\mathbf{r}| = r_1 + \cdots + r_d$. For computational feasibility, the infinite summation in Eq. (6) needs to be truncated at the finite term $P$. The truncated gPC expansion is given by,

$$x_i\left(t,\mathbf{Z}\right) = \sum_{|\mathbf{r}|=0}^{P} x_{i,\mathbf{r}}^{c}(t)\mathit{\Phi}_{\mathbf{r}}\left(\mathbf{Z}\right). \tag{7}$$

The total number of terms $(N+1)$ in gPC expansion of maximum polynomial order $P$ and $d$ uncertain variables is given by

$$N+1 = \frac{(d+P)!}{d!P!}. \tag{8}$$

One can map the vector-valued multi-index set $\mathbf{r}$ to the non-negative integers (using reverse-graded lexicographic technique in Ref. 31) and the resulting scalar indices correspond to different multi-dimensional basis functions. To that end, the gPC expansion in Eq. (7) with multidimensional index $\mathbf{r}$ can be written in terms of a single index $k$ as follows

$$x_i\left(t,\mathbf{Z}\right) = \sum_{k=0}^{N} x_{i,k}^{c}(t)\mathit{\Phi}_{k}\left(\mathbf{Z}\right) = \mathbf{\Phi}(\mathbf{Z})^{\mathrm{T}}\mathbf{x}_i^{c}(t) \tag{9}$$

where $\mathbf{x}_i^c(t) = [x_{i,0}^c(t), x_{i,1}^c(t),\ldots,x_{i,N}^c(t)] \in \mathbb{R}^{N+1}$ is the vector of time-varying gPC coefficients and $\mathbf{\Phi}(\mathbf{Z}) = [\mathit{\Phi}_0(\mathbf{Z}),\mathit{\Phi}_1(\mathbf{Z}),\ldots,\mathit{\Phi}_N(\mathbf{Z})]^{\mathrm{T}} \in \mathbb{R}^{N+1}$ is the vector of gPC basis. Now, using (9), we can write the gPC expanded solution of (5) in a compact form as follows

$$\mathbf{x}(t,\mathbf{Z}) = \left[\mathbf{I}_n \otimes \mathbf{\Phi}(\mathbf{Z})^{\mathrm{T}}\right]\mathbf{x}^c(t) \tag{10}$$

where $\mathbf{x}^c(t) = \mathrm{col}(\mathbf{x}_1^c(t), \mathbf{x}_2^c(t), \ldots, \mathbf{x}_n^c(t)) \in \mathbb{R}^{(N+1)n}$.

The gPC expansion requires a polynomial basis to be orthogonal with respect to the probability density function of the underlying random variable and satisfy the orthogonality relation in Eq. (2). The gPC expansion chooses the basis of expansion from the Askey scheme of polynomials (Ref. 32), which forms a complete basis in the Hilbert space determined by corresponding support (Ref. 33). Table 1 provides the correspondence of gPC orthogonal polynomials to different distributions of random variable $Z_i$ for $i = 1, \ldots, d$.

**Table 1.—**Link between type of gPC basis polynomials and their underlying random variables

| Distribution of $Z_i$ | Orthogonal polynomials | Support |
|---|---|---|
| Gaussian | Hermite | $Z_i \in (-\infty, \infty)$ |
| Gamma | Laguerre | $Z_i \in [0, \infty)$ |
| Beta | Jacobi | $Z_i \in$ [0,1] |
| Uniform | Legendre | $Z_i \in$ [-1,1] |

### 2.2.1 gPC expansion coefficients and Pseudospectral Stochastic Collocation

Let $\boldsymbol{\rho}(\mathbf{Z})$ represent the joint PDF of the random vector $\mathbf{Z}$ with support $\Gamma \subset \mathbb{R}^d$. Then, the coefficients of gPC expansion in Eq. (9) can be expressed as the following

$$x_{i,k}^c(t) = \frac{1}{\gamma_k}\mathbb{E}\left[x_i(t,\mathbf{Z})\Phi_k(\mathbf{Z})\right] = \frac{1}{\gamma_k}\int_\Gamma x_i(\mathbf{Z})\Phi_k(\mathbf{Z})\boldsymbol{\rho}(\mathbf{Z})d\mathbf{Z} \tag{11}$$

where $\gamma_k = \mathbb{E}\left[\Phi_k^2\right] > 0$ is the normalization factor. In the above Eq. (11), it is not trivial to exactly calculate the expectation term by solving for the integral under the PDF. Therefore, various numerical techniques are utilized in the literature to approximate the expectation integral and compute the gPC expansion coefficients. In this work, we utilize the approach of pseudospectral stochastic collocation proposed in Ref. 12 to compute the gPC expansion coefficients.

Stochastic collocation is the approach of generating an ensemble of solutions by solving the SDE in Eq. (5) for a finite number of realizations of the random vector $\mathbf{Z}$. These realizations can be obtained by randomly drawing independent samples from the given joint probability density function of the random vector or using deterministic nodes given by a specific quadrature technique. These nodal points are generally termed collocation points. At these sampled collocation nodes, the system of SDEs can be solved deterministically. Once the solution ensemble is generated, the idea behind the pseudospectral collocation approach is to approximate the expectation integral in Eq. (11) by a suitable multi-dimensional quadrature rule to compute the coefficients of gPC expansion.

Let us consider a multi-dimensional quadrature rule of $M$ nodes with $\{\mathbf{Z}_q\}$ as the set of quadrature nodes and $\{w_q\}$ as the corresponding set of nodal weights. Based on the numerical quadrature technique, the expansion coefficients in Eq. (11) can be approximated as the following

$$x_{i,k}^c(t) = \frac{1}{\gamma_k}\int_\Gamma x_i(t,\mathbf{Z})\;\boldsymbol{\Phi}_k(\mathbf{Z})\;\boldsymbol{\rho}(\mathbf{Z})\;d\mathbf{Z} \approx \frac{1}{\gamma_k}\sum_{q=1}^{M} x_i\left(\mathbf{Z}_q\right)\boldsymbol{\Phi}_k\left(\mathbf{Z}_q\right)w_q \tag{12}$$

where $x_i\left(t,\mathbf{Z}_q\right)$, $i = 1,\ldots,n$ is the solution of the SDE in Eq. (5) at the quadrature node $\mathbf{Z}_q$. Note that the convergence of the solution obtained from the pseudospectral approach is completely dependent on the convergence behavior of the quadrature-based integral approximation.

Let us define $\bar{\mathbf{x}}_i(t) = \left[x_i\left(t, \mathbf{Z}^{(1)}\right), x_i\left(t, \mathbf{Z}^{(2)}\right), \cdots, x_i\left(t, \mathbf{Z}^{(N_q)}\right)\right]^{\mathrm{T}} \in \mathbb{R}^{N_q}$ and $\bar{\mathbf{x}}(t) = \mathrm{col}(\bar{\mathbf{x}}_1(t), \bar{\mathbf{x}}_2(t), \ldots, \bar{\mathbf{x}}_n(t)) \in \mathbb{R}^{nN_q}$. Using Eq. (12), we can obtain the gPC coefficient $\mathbf{x}^c(t)$ in Eq. (10) as follows

$$\mathbf{x}^{\mathrm{c}}(t) = \left[\mathbb{I}_n \otimes \mathbf{\Gamma}^{-1}\right] [\mathbf{I}_n \otimes \bar{\mathbf{\Phi}}][\mathbf{I}_n \otimes \mathbf{W}]\bar{\mathbf{x}}(t) \tag{13}$$

where $\mathbf{W} = \mathrm{diag}(w_1, \cdots, w_{N_q}) \in \mathbb{R}^{N_q \times N_q}$ is the diagonal matrix of quadrature weights and the matrix $\bar{\mathbf{\Phi}} \in \mathbb{R}^{(N+1) \times N_q}$ is defined as follows

$$\bar{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi}_0\left(\mathbf{Z}^{(1)}\right) & \mathbf{\Phi}_0\left(\mathbf{Z}^{(2)}\right) & \cdots & \mathbf{\Phi}_0\left(\mathbf{Z}^{(N_q)}\right) \\ \mathbf{\Phi}_1\left(\mathbf{Z}^{(1)}\right) & \mathbf{\Phi}_1\left(\mathbf{Z}^{(2)}\right) & \cdots & \mathbf{\Phi}_1\left(\mathbf{Z}^{(N_q)}\right) \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{\Phi}_N\left(\mathbf{Z}^{(1)}\right) & \mathbf{\Phi}_N\left(\mathbf{Z}^{(2)}\right) & \cdots & \mathbf{\Phi}_N\left(\mathbf{Z}^{(N_q)}\right) \end{bmatrix}. \tag{14}$$

Using the Kronecker identities, we can express Eq. (13) as

$$\mathbf{x}^c(t) = \left[\mathbf{I}_n \otimes \mathbf{\Gamma}^{-1}\bar{\mathbf{\Phi}}\mathbf{W}\right] \bar{\mathbf{x}}(t). \tag{15}$$

### 2.2.2 Statistics of the Solution using gPC Expansion

Once the coefficients of the gPC expansion are computed using Eq. (15), various statistical measures of the solution of the stochastic system in Eq. (5), $x(t, \mathbf{Z})$, can be computed. Note that the gPC polynomials are orthogonal such that $\mathbb{E}[\mathit{\Phi}_i(\mathbf{Z})\mathit{\Phi}_j(\mathbf{Z})] = 0$ for all $i \neq j$ and $\mathbb{E}[\mathit{\Phi}_i^2(\mathbf{Z})] = \gamma_i$. To that end, one can write the following $\mathbb{E}\left[x_i(t, \mathbf{Z})\right] \approx x_{i,0}^c(t)$. Therefore, the mean vector of the solution of Eq. (5) can be written as

$$\boldsymbol{\mu}_{\mathbf{x}}(t) = \mathbb{E}\left[\mathbf{x}(t, \mathbf{Z})\right] \approx \mathbf{x}^{c_0} \tag{16}$$

where $\mathbf{x}^{c_0} = [x_{1,0}^c, x_{2,0}^c, \ldots, x_{n,0}^c] \in \mathbb{R}^n$. Further, we can compute the correlation matrix $\mathbf{R}_{\mathbf{xx}} = \mathbf{R}_{\mathbf{xx}}^{\mathrm{T}} \in \mathbb{R}^{n \times n}$ as

$$\mathbf{R}_{\mathbf{xx}}(t) = \mathbb{E}[\mathbf{x}(t, \mathbf{Z})\mathbf{x}(t, \mathbf{z})^{\mathrm{T}}] \approx \mathbf{x}^c(t) \left(\mathbf{I}_n \otimes \mathbf{\Gamma}\right) \mathbf{x}^c(t)^{\mathrm{T}}. \tag{17}$$

The covariance matrix can be computed as follows

$$\mathbf{P}_{\mathbf{xx}}(t) = \mathbf{R}_{\mathbf{xx}}(t) - \boldsymbol{\mu}_{\mathbf{x}}(t)\boldsymbol{\mu}_{\mathbf{x}}(t)^{\mathrm{T}}. \tag{18}$$

Given another stochastic process $\mathbf{y}(t, \mathbf{Z}) \in \mathbb{R}^{m \times m}$, the cross-correlation matrix $\mathbf{R}_{\mathbf{xy}} \in \mathbb{R}^{n \times m}$ can be computed as follows

$$\mathbf{R}_{\mathbf{xy}}(t) = \mathbb{E}[\mathbf{x}(t, \mathbf{Z})\mathbf{y}(t, \mathbf{Z})^{\mathrm{T}}] \approx \mathbf{x}^c(t) \left(\mathbf{I}_n \otimes \mathbf{\Gamma}\right) \mathbf{y}^c(t)^{\mathrm{T}} \tag{19}$$

where $\mathbf{y}^c \in \mathbb{R}^{(N+1)m}$ is the $N$th order gPC coefficient vector of the stochastic process $\mathbf{y}(t, \mathbf{Z})$. Similarly, the cross-covariance matrix can be computed as follows

$$\mathbf{P}_{\mathbf{xy}}(t) = \mathbf{R}_{\mathbf{xy}}(t) - \boldsymbol{\mu}_{\mathbf{xy}}(t)\boldsymbol{\mu}_{\mathbf{xy}}(t)^{\mathrm{T}} \tag{20}$$

where $\boldsymbol{\mu}_{\mathbf{xy}}(t) = [\boldsymbol{\mu}_{\mathbf{x}}^{\mathrm{T}}(t), \boldsymbol{\mu}_{\mathbf{y}}^{\mathrm{T}}(t)]^{\mathrm{T}}$ with $\boldsymbol{\mu}_{\mathbf{y}}(t) = \mathbb{E}\left[\mathbf{y}(t, \mathbf{Z})\right] \approx \mathbf{y}^{c_0}$ as the mean of the stochastic process $\mathbf{y}(t, \mathbf{Z})$.

### 2.2.3 gPC Expansion in Systems with Different PDF Types and Mixed Sparse Grid Quadrature

Note that the SDE in Eq. (5) is a function of random vector $\mathbf{Z} = [Z_1, \ldots, Z_d]$ whose joint PDF is a product of different marginal PDF types. Let the marginal PDF of the random variable $Z_i$, $i = 1, \ldots, d$ be denoted by $\rho_{z_i}$. Since the random variables are independent, the joint probability distribution is given by,

$$\rho_{\mathbf{Z}} = \rho_{Z_1} \times \ \rho_{Z_2} \times \ldots \times \ \rho_{Z_m} \tag{21}$$

where, $\rho_{\mathbf{Z}}$ is the joint probability distribution of the random vector $\mathbf{Z}$. As stated earlier in Section 2.1, we consider the random vector $\mathbf{Z}$ in the form, $\mathbf{Z} = [\mathbf{Z}_{x_0}^{\mathrm{T}}, \mathbf{Z}_p^{\mathrm{T}}]^{\mathrm{T}}$ where

$$\begin{aligned} \mathbf{Z}_{\mathbf{x}_{0_i}} &\sim \mathcal{N}(\mu_i, \sigma_i^2) \text{ for } i = 1, \ldots, n \\ \mathbf{Z}_{\mathbf{p}_j} &\sim \mathcal{U}[a_j, b_j] \text{ for } j = 1, \ldots, n_p \end{aligned} \tag{22}$$

Further, to carry out a gPC expansion for the random process where the PDF of the random variables are of mixed types as in Eq. (22), it is required to construct a polynomial basis which is orthogonal with respect to the mixed joint PDF $\rho_{\mathbf{Z}}$. To that end, the multivariate gPC basis orthogonal to the mixed joint PDF can be obtained as follows

$$\Phi_{\mathbf{r}}(\mathbf{Z}) = \prod_{j=1}^{d} \Phi_{r_j}(Z_j) \tag{23}$$

where $\Phi_{r_j}(Z_j)$, $j = 1, \ldots, d$ is the $j$th univariate basis in random variable $Z_j$ , and $\mathbf{r} = (r_1, \ldots, r_d)$ is the ordered set of multi-indices with $|\mathbf{r}| = r_1 + \cdots + r_d$. Note that for the PDF of random variables as in Eq. (22), from Table 1, $\Phi_{r_i}(Z_i)$, $i = 1, \ldots, n$ are the basis of Hermite polynomials and $\Phi_{r_j}(Z_j)$, $j = 1, \ldots, n_p$ are the basis of Legendre polynomials.

To compute the coefficients of the gPC expansion using pseudospectral collocation for the problems with mixed PDF types, we consider the mixed sparse grid quadrature technique-based collocation nodes proposed in Ref. 13. In general, a conventional sparse grid uses a linear combination of lower-dimensional tensor products of the Gaussian quadrature rules to generate the sparse grid-based collocation nodes. The sparse grid-based tensor product is carried out using isotropic 1-D quadrature rules in all dimensions. For example, for a problem involving two random variables $Z_1$ and $Z_2$ governed by the Gaussian density function, the conventional sparse grid quadrature nodes and weights are generated by considering the sparse tensor product of Gaussian Hermite quadrature on both the dimensions. Because of this isotropy, the conventional sparse grid quadrature can provide the collocation nodes with random variables governed by a single distribution function (either uniform or normal) only. On the other hand, the mixed sparse grid (MSG) can be generated by using a mixture of 1-D quadrature rules in different dimensions depending on the distribution of the random variables. For further details on MSG quadrature, please refer to Ref. 13. Refer to Ref. 34 for the algorithm to generate the sparse grid nodes and weights.

### 2.2.4 gPC Expansion-based Statistics

Given the estimates of the gPC coefficients, the approximate statistics i.e. mean and variance of $x_i(\mathbf{Z})$ can be computed as,

$$\begin{aligned} \mathbb{E}\left[x_i(t, \mathbf{Z})\right] &\approx \mathbb{E}\left[\sum_{k=0}^{N} x_{i,k}^c(t) \Phi_k(\mathbf{Z})\right] = x_{i,0}^c(t) \\ \text{var}\left[x_i(t, \mathbf{Z})\right] &\approx \mathbb{E}\left[\left(\sum_{k=0}^{N} x_{i,k}^c(t) \Phi_k(\mathbf{Z}) - \mathbb{E}\left[x_i(t, \mathbf{Z})\right]\right)^2\right] = \sum_{k=1}^{N} \gamma_k \left(x_{i,k}^c(t)\right)^2 \end{aligned} \tag{24}$$

where $\mathbb{E}[\cdot]$ represents the expectation operator to compute mean and var$[\cdot]$ denotes the variance operator. One can also use the gPC expansion technique to compute the numerical PDF of $x_i(\mathbf{Z})$ by evaluating the gPC expansion expression in Eq. (9) at a large number of samples of random vector $\mathbf{Z}$.

### 2.2.5 gPC Expansion-Based Sensitivity Analysis

Sensitivity analysis aids us to make informed decisions about the sensitivity of the system to the parameters involved. Among all the variables, some of the variables have a significant influence on the response of the system, and these parameters need to be modeled more precisely than the others. Polynomial chaos-based Sobol' indices for sensitivity analysis were introduced by Sudret Ref. 21.

In this section, we consider the sensitivity analysis of a quantity of interest given by the function $g(x_i, t)$ where $x_i$, $i = 1, \dots, n$ are the state variables of the SDE in Eq. (5). The first step of carrying out sensitivity analysis is to approximate the function $g(x_i, \mathbf{Z}, t)$ by gPC expansion of suitable order as follows

$$g\left(x_i, t, \mathbf{Z}\right) = \sum_{|\mathbf{r}|=0}^{P} g^c_{\mathbf{r}}(x_i, t) \mathit{\Phi}_{\mathbf{r}}\left(\mathbf{Z}\right). \tag{25}$$

where $g^c_{\mathbf{r}}(t)$ is the coefficient of the multidimensional basis $\mathit{\Phi}_{\mathbf{r}}\left(\mathbf{Z}\right)$ and $\mathbf{r} = (r_1, \dots, r_d)$ is the ordered set of multi-indices with $|\mathbf{r}| = r_1 + \cdots + r_d$. For a given gPC expansion, the Sobol' indices at any order may be obtained by the combination of the squares of the suitable gPC coefficients $g^c_{\mathbf{r}}(t)$. The total sensitivity index $\mathcal{S}_j(t)$ of the function $g(x_i, \mathbf{Z}, t)$ with respect to the random variable $Z_j$, $j = 1, \dots, d$ is given by,

$$\mathcal{S}_j(t) = \frac{\sum_{\alpha \in \mathcal{I}_j} g^{c^2}_{\alpha}\left(x_i, t\right) \mathbb{E}\left[\mathbf{\Phi}^2_{\alpha}\right]}{\text{var}\left[g(x_i, t, \mathbf{Z})\right]} \tag{26}$$

where, $\mathcal{I}_j = \{\mathbf{r} \ | \ r_j > 0\}$ corresponds to the polynomials depending on random input $Z_i$ and possibly on other random inputs as well. Further, the variance of the function var$\left[g\left(x_i, t, \mathbf{Z}\right)\right]$ can be computed using the Eq. (24).

# 3 Vehicle Models and Autopilots

## 3.1 Equations of Motion of Fixed-wing sUAVs

We consider the following fixed-wing sUAV model, augmented with equations of motion for altitude, specific energy, flight path angle, and linear first-order lateral and longitudinal actuator dynamics (Ref. 35), where $X$ and $Y$ are the north and east coordinates of the sUAV, respectively, $H$ is the altitude of the sUAV from the origin of the inertial frame, $E$ is the specific energy and $\gamma$ and $\chi$ are the flight path angle and heading angle, respectively as illustrated in Fig. 2. Further, $\eta$, $a_p$, and $a_y$ are the throttle position, pitch acceleration, and yaw acceleration, respectively, while $\eta_c$, $a_{pc}$, and $a_{yc}$ are the commanded throttle, pitch acceleration, and yaw acceleration, respectively. Here, $\lambda_n$, $\lambda_p$, and $\lambda_y$ are the actuator time constants corresponding to throttle, pitch, and yaw.

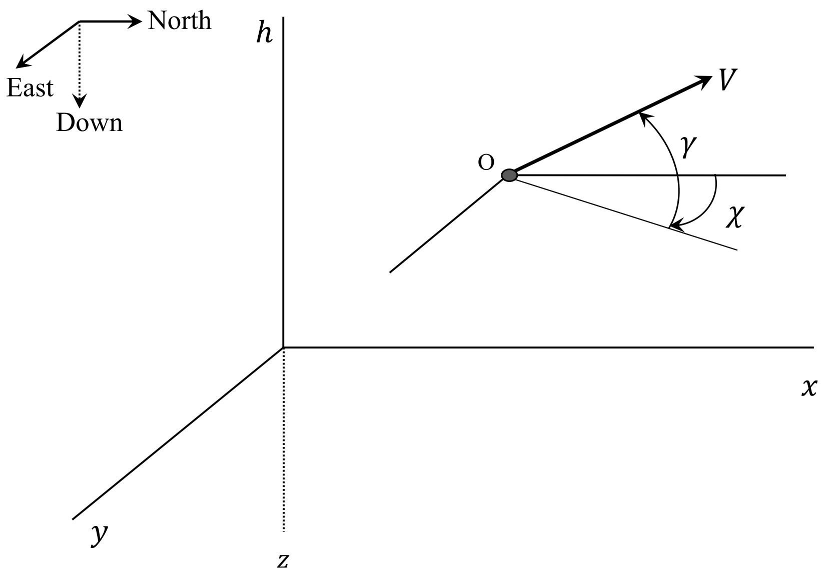


**Figure 2.**—Three dimensional flight of sUAV

$$
\begin{aligned}
\dot{X} &= V\cos\gamma\cos\chi \\
\dot{Y} &= V\cos\gamma\sin\chi \\
\dot{H} &= V\sin\gamma \\
\dot{E} &= \frac{V}{W}(T-D) \\
\dot{\gamma} &= \frac{1}{V}(a_p - g\cos\gamma) \\
\dot{\chi} &= \frac{a_y}{V\cos\gamma} \\
\dot{\eta} &= -\lambda_n(\eta - \eta_c) \\
\dot{a}_p &= -\lambda_p(a_p - a_{pc}) \\
\dot{a}_y &= -\lambda_y(a_y - a_{yc})
\end{aligned}
\tag{27}
$$

It is assumed that the thrust is in the direction of the velocity of the sUAV and is calculated as follows

$$T = \eta T_{\text{max}} \tag{28}$$

where $0 < \eta < 1$. The total drag acting on the sUAV is assumed to be the sum of parasite drag, $D_0$ and induced drag, $D_i$ and is given as follows

$$D = D_0 + n^2 D_i, \quad D_0 = qSC_{D0}, \quad D_i = k\left(W/qS\right)^2 \tag{29}$$

where $q = \frac{1}{2}\rho V^2$ is the dynamic pressure, $\rho$ is the atmospheric density, $S$ is the wing planform area, $C_{D0}$ is the zero-lift drag coefficient, $k$ is the induced drag parameter, $W$ is the weight, and $n$ is the load-factor of the sUAV. The velocity $V$ of the sUAV is given by

$$V = \sqrt{2g\,(E - H)} \tag{30}$$

In the cases when the UAVs are flying through a non-constant wind flow-field $\mathbf{W} = [W_x, W_y, 0]^{\mathrm{T}}$, they are influenced by the components of the wind field, local temporal variations of the wind field and gradients of the wind field. With these premises, we augment the equations described above as (see Ref. 36 and Ref. 37):

$$\begin{aligned}
\dot{x} &= V\cos\chi + W_x \\
\dot{y} &= V\sin\chi + W_y \\
\dot{V} &= \frac{g}{W}(T - D) - \dot{W}_x\cos\chi - \dot{W}_y\sin\chi \\
\dot{\chi} &= \left(\frac{g}{V}\right) n\sin\mu + \frac{1}{V}\left(\dot{W}_x\sin\chi - \dot{W}_y\cos\chi\right)
\end{aligned} \tag{31}$$

where $x$ and $y$ are the inertial North and East coordinates of the UAV. Here, $V$ denotes the velocity of the UAV relative to the air. The heading angle of the UAV is denoted by $\chi$ and $\mu$ is the bank angle. Further, $L$, $D$, and $T$ are the lift, drag, and thrust, respectively, acting on the UAV.

We further assume that the wind conditions change slowly with time at a point in the trajectory (steady wind flow-field). With this assumption, the total derivative of the wind velocity components can be written as follows

$$\begin{aligned}
\dot{W}_x &= \left(\frac{\partial W_x}{\partial x}\right)\dot{x} + \left(\frac{\partial W_x}{\partial y}\right)\dot{y} \\
\dot{W}_y &= \left(\frac{\partial W_y}{\partial x}\right)\dot{x} + \left(\frac{\partial W_y}{\partial y}\right)\dot{y}.
\end{aligned} \tag{32}$$

where the partial derivatives are the spatial gradients of wind velocity.

We consider the following two control variables: (i) the throttle Position, $\eta$, (ii) the bank Angle, $\mu$ in this section to steer the vehicles along a filed path. To that end, we consider the following first-order actuator dynamics.

$$\begin{aligned}
\dot{\eta} &= -\lambda_\eta(\eta - \eta_c), \quad \lambda_\eta = \frac{1}{\tau_\eta} \\
\dot{\mu} &= -\lambda_\mu(\mu - \mu_c), \quad \lambda_\mu = \frac{1}{\tau_\mu}
\end{aligned} \tag{33}$$

where $\tau_\eta$ and $\tau_\mu$ are the actuator time constants. In Eq. (33), $0 \le \eta_c \le 1$ and $\mu_c$ denote the throttle position and bank angle commands. The load factor $n$ is bounded by the structural limit and satisfies

$$n \le n_{\text{max}}. \tag{34}$$

Additionally, the load factor is bounded by the maximum lift coefficient $C_{L_{\text{max}}}$ such that

$$n \le n_L = \frac{qS}{W} C_{L_{\text{max}}}. \tag{35}$$

The bound on the load factor leads to a bound on the bank angle Ref. 36. Note that for turning flight in a horizontal plane, $L\cos\mu = W$. Therefore, we bound the bank angle command as follows

$$|\mu_c| \le \arccos\left(\frac{1}{n_{\text{max}}}\right). \tag{36}$$

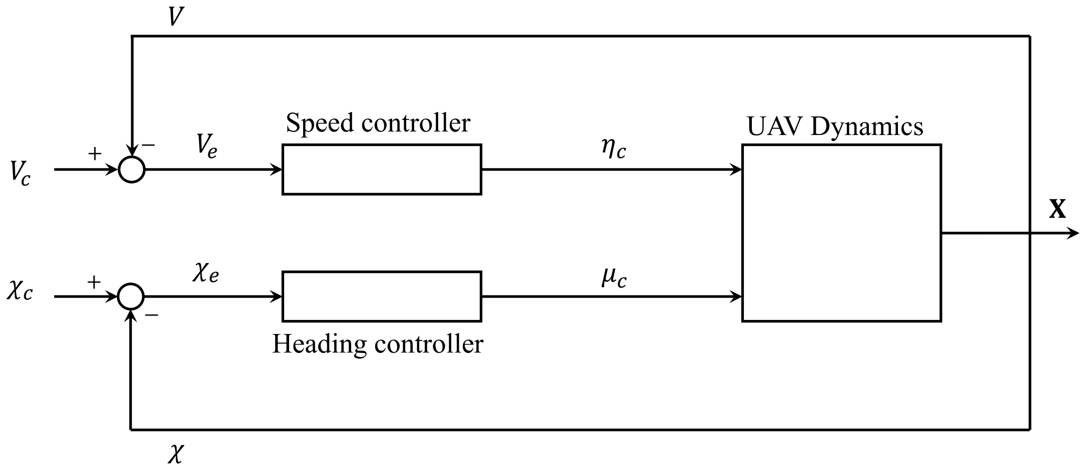


**Figure 3.—**Configuration of the PI autoilot

## 3.2 Speed and Heading Autopilot for Fixed-Wing UAVs

Here, we design a proportional-integral (PI) speed and heading controller for the fixed wing UAV. The autopilot configuration is depicted in Fig. 3. Let $V_c$ and $\chi_c$ denote the commanded velocity and heading angle of a UAV in the airspace. Then, the throttle position and bank angle commands are generated using the following:

$$\begin{aligned} \eta_c =& \quad K_{V_P} V_e + \left( \frac{K_{V_I}}{s} \right) V_e \\ \mu_c =& \quad K_{\mu_P} \chi_e + \left( \frac{K_{\mu_I}}{s} \right) \chi_e. \end{aligned} \tag{37}$$

where $V_e = V_c - V$ and $\chi_e = \chi_c - \chi$ are the error terms in speed and heading angle.

## 3.3 Equations of Motion of Quadrotor in a Windshear

Similar to that with fixed-wing UAVs, we assume that the quadrotors are flying through a non-constant wind flow-field $\mathbf{W} = [W_x, W_y, 0]^{\mathrm{T}}$. The governing dynamics of the quadrotor in a steady wind flow-field are modeled as follows

$$\begin{aligned} \dot{x} &= \dot{x} + W_x \\ \dot{y} &= \dot{y} + W_y \\ \ddot{x} &= \frac{u_1}{m} (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) - \dot{W}_x \\ \ddot{y} &= \frac{u_1}{m} (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) - \dot{W}_y \\ \ddot{z} &= \frac{u_1}{m} (\cos\phi \cos\theta) - g \end{aligned}$$

$$
\begin{aligned}
\ddot{\phi} &= \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_r}{I_x}\dot{\theta}\Omega_r + \frac{L}{I_x}u_2 \\
\ddot{\theta} &= \dot{\psi}\dot{\phi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_r}{I_y}\dot{\phi}\Omega_r + \frac{L}{I_y}u_3 \\
\ddot{\psi} &= \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}u_4
\end{aligned}
\tag{38}
$$

where $x$, $y$, and $z$ are the position coordinates of the quadrotor along North, East, and Down axis-frames. Here, $\phi$, $\theta$, and $\psi$ are the Euler angles of the quadrotor in roll, pitch, and yaw. $I_x$, $I_y$, and $I_z$ are the principal moments of inertia of the quadrotor. $J_r$ is the inertia of rotor, $L$ is the arm length, $b$ is the thrust constant, and $d$ is the drag factor. The control inputs in terms of rotor speeds are as follows

$$
\begin{aligned}
u_1 &= b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\
u_2 &= b(\omega_4^2 - \omega_2^2) \\
u_3 &= b(\omega_3^2 - \omega_1^2) \\
u_4 &= d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)
\end{aligned}
\tag{39}
$$

where $\omega_i$, $i = 1, 2, 3, 4$ are the rotor speeds and the relative rotor-speed is given by

$$
\Omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4 \tag{40}
$$

### 3.4 Autopilot for Quadrotor

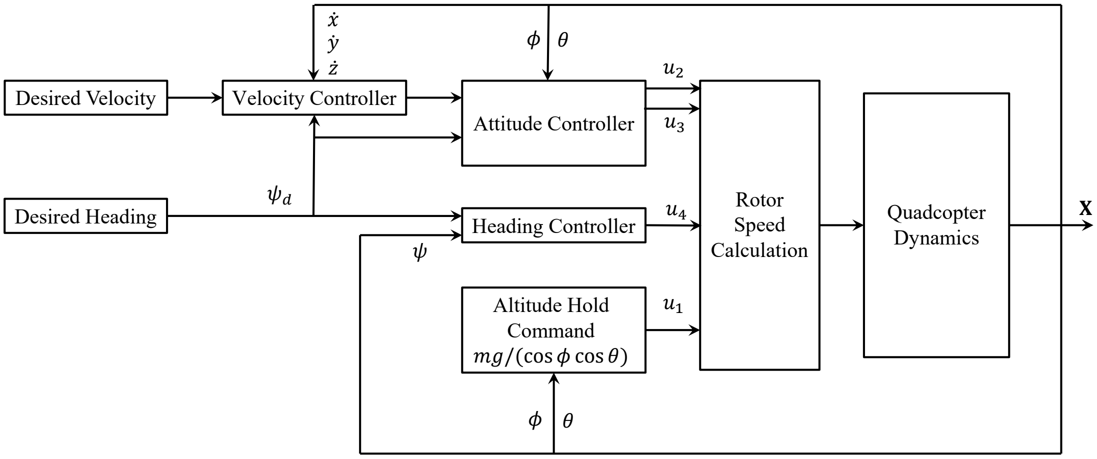


**Figure 4.—**Autopilot for quadrotor

Figure 4 illustrates the autopilot configuration for the quadrotor. Similar to fixed-wing UAVs, the motion of the quadrotors are restricted on a horizontal plane by considering the following control command

$$
u_1 = \frac{mg}{\cos(\phi)\cos(\theta)}
$$

We consider a proportional (P) speed controller to track the commanded forward speed of the quadrotor. Further, we consider a proportional-derivative (PD) heading controller to track the commanded heading. The commanded heading and the speed of the quadrotor are used to generate pitch and roll commands. Finally, we design PD attitude controllers to track the generated pitch and roll commands. These controllers generate the desired control inputs $u_i$, $i = 1, 2, 3, 4$ which are then used to compute the rotor speeds $\omega_i$, $i = 1, 2, 3, 4$ and the relative rotor-speed $\Omega_r$.

# 4 Uncertainty Quantification and Conflict Assessment in a Traffic of Fixed-Wing UAVs

In this section, we carry out uncertainty propagation in traffic of fixed-wing small unmanned aerial vehicles (sUAVs) subject to vehicle uncertainties and assess the risk of conflict in the traffic. The uncertainty propagation is carried out using a pseudospectral collocation-based gPC expansion technique.

## 4.1 Preliminary Assessment with two sUAVs in the Airspace

### 4.1.1 Trajectory generation and distribution of uncertainties

We consider a filed flight plan with wing level, constant-altitude motion for both the sUAVs (i.e. $\dot{H} = 0$ and $\dot{\gamma} = 0$) which can be attained by enforcing $a_p(0) = g$, $a_{pc} = g$, and $\gamma(0) = 0$. Also, the flight plan is generated with a constant commanded yaw acceleration of $a_{yc} = 0.01g$. Further, we assume uncertainties in the initial position coordinates ($X$ and $Y$) of both the sUAVs. The distribution of uncertainties in the initial position coordinates is illustrated in Table 2.

**Table 2.—**Initial condition uncertainties

| Initial Position along | Distribution Type | PDF |
|---|---|---|
| North (sUAV-1) | Normal | $\mathcal{N}(0, 0.01^2)$ m |
| East (sUAV-1) | Normal | $\mathcal{N}(0, 0.01^2)$ m |
| North (sUAV-2) | Normal | $\mathcal{N}(2000, 1^2)$ m |
| East (sUAV-2) | Normal | $\mathcal{N}(500, 1^2)$ m |

Both the sUAVs are assumed to have the same initial altitude $H(0) = 250$ m, specific energy $E(0) = 270$ m, flight path angle, $\gamma(0) = 0$ rad, throttle position $\eta(0) = 1$, pitch acceleration $a_p(0) = g$, and yaw acceleration $a_y(0) = 0.01g$. The initial heading of the sUAV-1 is assumed to be $\chi(0) = 0$ radians and that of sUAV-2 is assumed to be $\chi(0) = 0.15$ radians.

In addition to the initial condition uncertainties, we assume uncertainties in the weight, zero-lift drag coefficient, wing planform area, induced drag parameter, and maximum available thrust. The parametric uncertainties are assumed to be the same for both sUAVs. Table 3 illustrates the distribution of uncertainties in the parameters of the sUAVs. Other simulation parameters are assumed to be deterministic and equal for both the sUAVs as follows: $\lambda_n = \lambda_p = \lambda_y = 0.15$, $g = 9.81$ m/s$^2$, and $\rho = 1.225$ Kg/m$^3$. We employ a $4^{th}$ order (i.e. $P = 4$) gPC expansion technique to carry out uncertainty propagation.

### 4.1.2 Simulation Results

Figure 5 illustrates the temporal variation of the mean and $3 - \sigma$ confidence ellipses of the sUAVs' position along north and east. Further, Fig. 6 depicts the mean and $3 - \sigma$ confidence ellipses of the sUAVs'

**Table 3.—**Parametric uncertainties

| Variables | Distribution Type | PDF |
|---|---|---|
| $m$ | Uniform | $\mathcal{U}[3, 4]$ Kgs |
| $C_{D0}$ | Uniform | $\mathcal{U}[0.03, 0.035]$ |
| $S$ | Uniform | $\mathcal{U}[0.5, 0.55]$ m$^2$ |
| $k$ | Uniform | $\mathcal{U}[0.015, 0.025]$ |
| $T_{\text{max}}$ | Uniform | $\mathcal{U}[5, 6]$ N |

position along north and east at various time instants. Clearly, upon analyzing Figs. 5 and 6, there is no risk of conflict among the two sUAVs during the flight time.

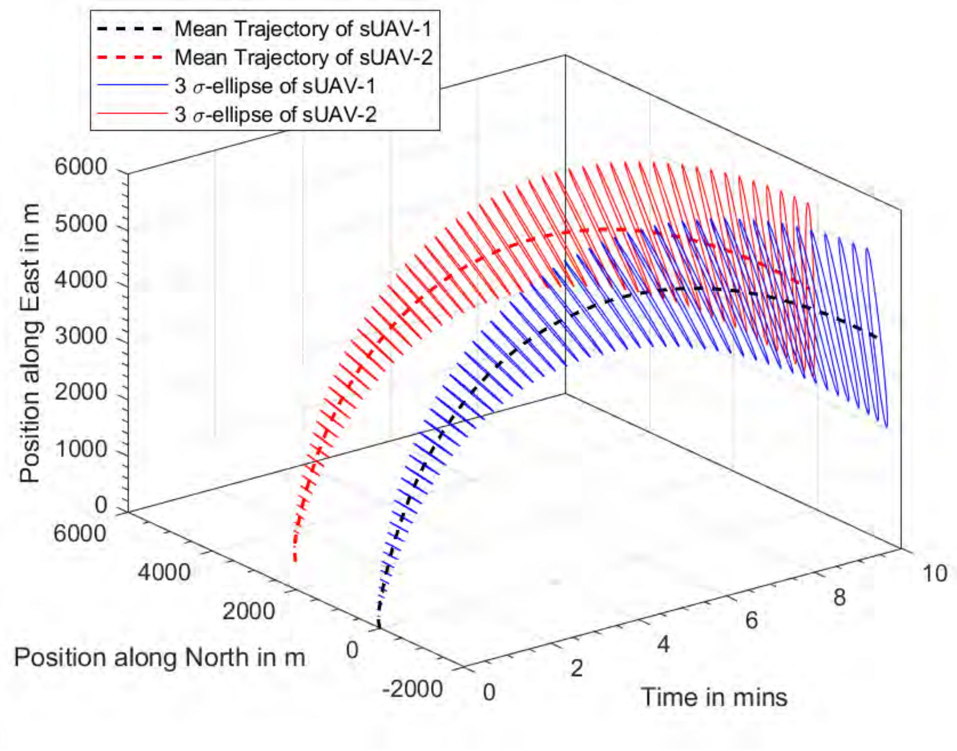


**Figure 5.—**Mean trajectory and 3-$\sigma$ confidence ellipses of sUAVs' positions

### 4.1.3 Sensitivity Analysis

Further, we analyze the sensitivity of a safety metric between the sUAVs to the underlying uncertainties using the framework discussed in Section 2.2. We consider the separation between the sUAVs, $d_{\text{sep}}(t)$ as the safety metric, which can be modeled as follows

$$d_{\text{sep}}(t) = \sqrt{(X_1(t) - X_2(t))^2 + (Y_1(t) - Y_2(t))^2} \tag{41}$$

where $X_1(t)$ and $X_2(t)$ are the positions of the sUAVs along the north, and $Y_1(t)$ and $Y_2(t)$ are the positions of the sUAVs along the east, respectively.

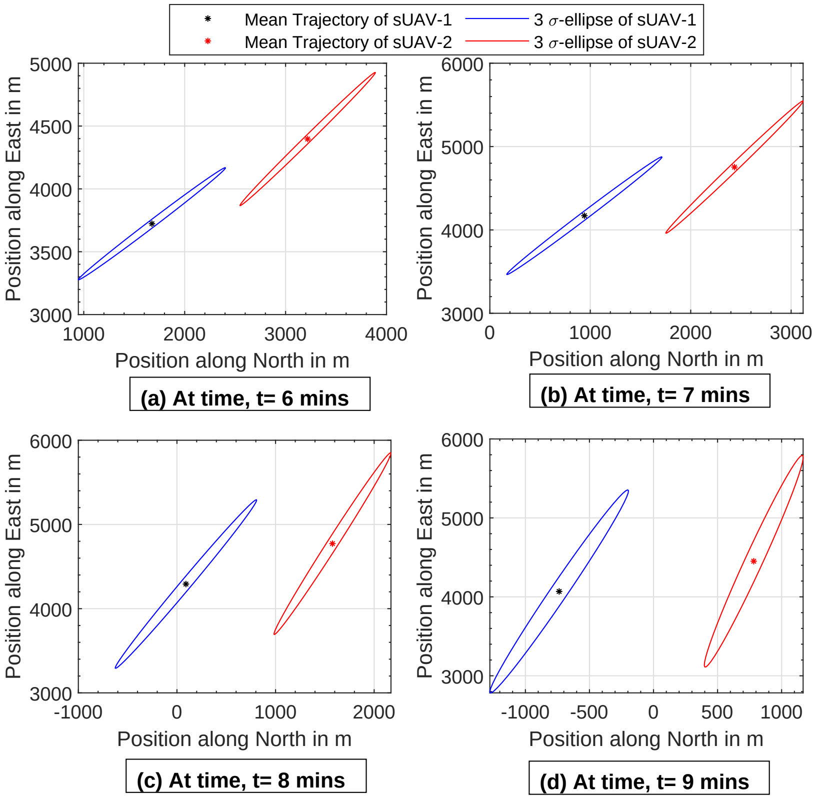


**Figure 6.—**Mean and 3-$\sigma$ confidence ellipses of sUAVs' positions at various time instants

Figure 7 depicts the temporal variation of the gPC expansion-based sensitivity indices of the safety metric with respect to various uncertain variables (mass, $m$; maximum thrust, $T_{max}$; Zero-lift drag coefficient, $C_{D0}$; Wing planform area, $S$; Initial position, North $N$, and East $E$). From Fig. 7, we see the separation between the 2 sUAVs (risk of conflict) is most influenced by:

(i) Maximum available thrusts ($T_{\max_1}$ and $T_{\max_2}$)

(ii) Zero-lift drag coefficients ($C_{D0_1}$ and $C_{D0_2}$)

(iii) Wing planform areas ($S_1$ and $S_2$)

### 4.1.4 Risk Analysis with larger uncertainties

To demonstrate the significance of sensitivity analysis and uncertainty quantification, in general, we carry out the simulations with larger values of uncertainties in the variables which most influence the safety metric. Table 4 illustrates the increased uncertainty bounds in the uncertain variables significant to the safety metric.

Figure 8 illustrates the temporal variation of the mean and $3-\sigma$ confidence ellipses of the sUAVs' position along north and east with larger uncertainties. Further, Fig. 9 depicts the mean and $3-\sigma$ confidence ellipses of the sUAVs' position along north and east at various time instants. Clearly, from Fig. 9, one can observe the colliding confidence ellipses in the two sUAVs' positions, which depicts the increased risk of conflict among the two sUAVs during the flight time. From Fig. 7 we note that the maximum thrust ($T_{max}$) is

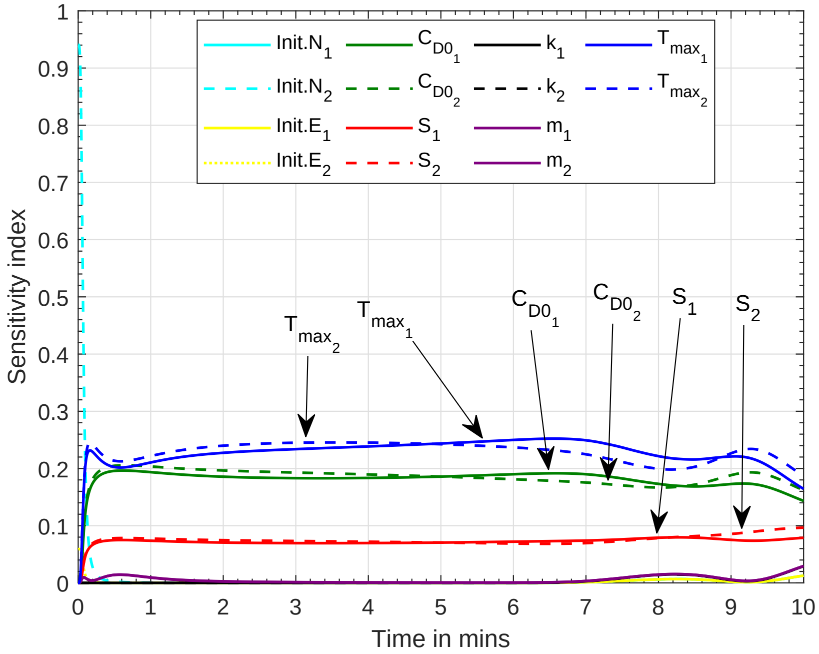


**Figure 7.—**Sensitivity of separation distance between sUAVs to various uncertainties

**Table 4.—**Increased uncertainty bounds for significant uncertain variables

| Variables | Distribution Type | PDF |
|---|---|---|
| $C_{D0}$ | Uniform | $\mathcal{U}[0.03, 0.06]$ |
| $S$ | Uniform | $\mathcal{U}[0.5, 0.8]$ m$^2$ |
| $T_{\max}$ | Uniform | $\mathcal{U}[5, 8]$ N |

the most influential parameter, followed by the zero-lift drag coefficient ($C_{D0}$), and then the wing planform area ($S$).

## 4.2 Conflict Risk Assessment in traffic of four sUAVs

### 4.2.1 Trajectory generation and distribution of uncertainties

Similar to the case with two sUAVs in the airspace, to conduct uncertainty quantification and conflict risk assessment, we consider wing level, constant-altitude motion for both the sUAVs (i.e. $\dot{H} = 0$ and $\dot{\gamma} = 0$). Further, instead of assuming constant commanded yaw acceleration, we obtained the filed flight path plan with different commanded yaw acceleration profiles for all 4 sUAVs as illustrated in Table 5. We assume uncertainties in the initial position coordinates ($X$ and $Y$) of all four sUAVs. The distribution of uncertainties in the initial position coordinates is illustrated in Table 6.

All four sUAVs are assumed to have same initial altitude $H(0) = 250$ m, specific energy $E(0) = 270$ m, flight path angle, $\gamma(0) = 0$ rad, throttle position $\eta(0) = 1$, pitch acceleration $a_p(0) = g$, and yaw acceleration $a_y(0) = 0$. The parametric uncertainties are assumed to be the same for all sUAVs as illustrated in Table 7. Other simulation parameters are assumed to be deterministic and equal for both the sUAVs as follows: $\lambda_n = \lambda_p = \lambda_y = 0.15$, $g = 9.81$ m/s$^2$, and $\rho = 1.225$ Kg/m$^3$. We employ a $4^{th}$ order (i.e. $P = 4$)

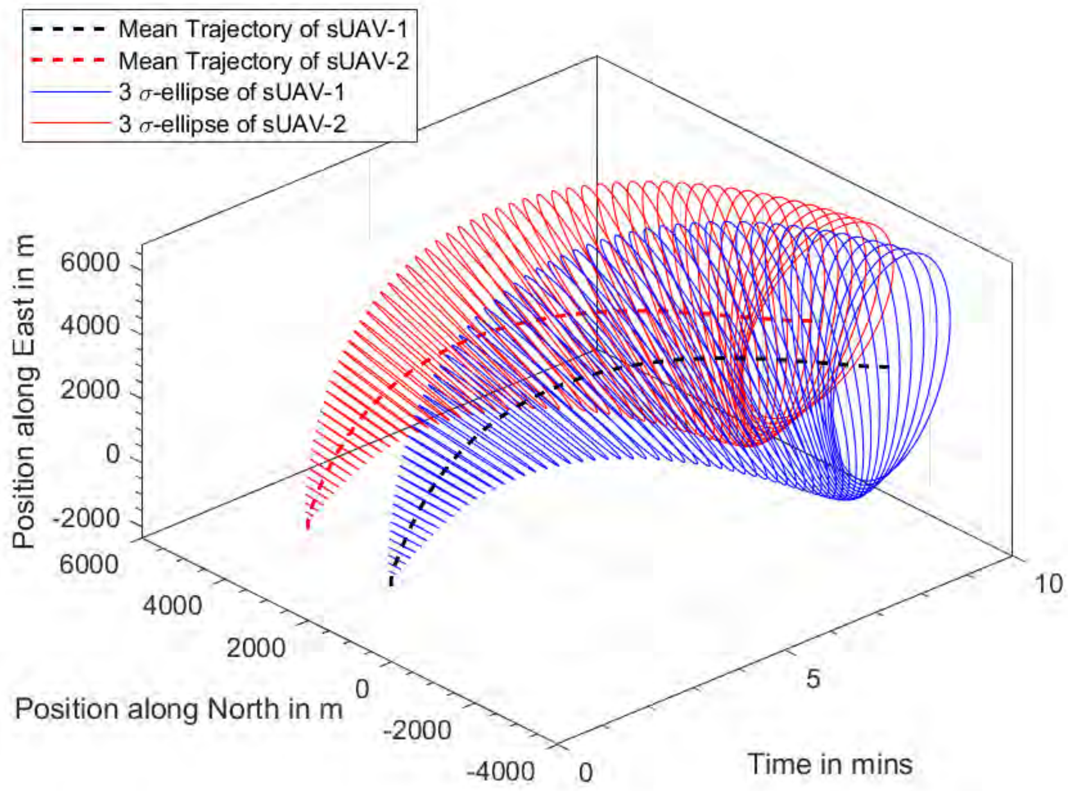
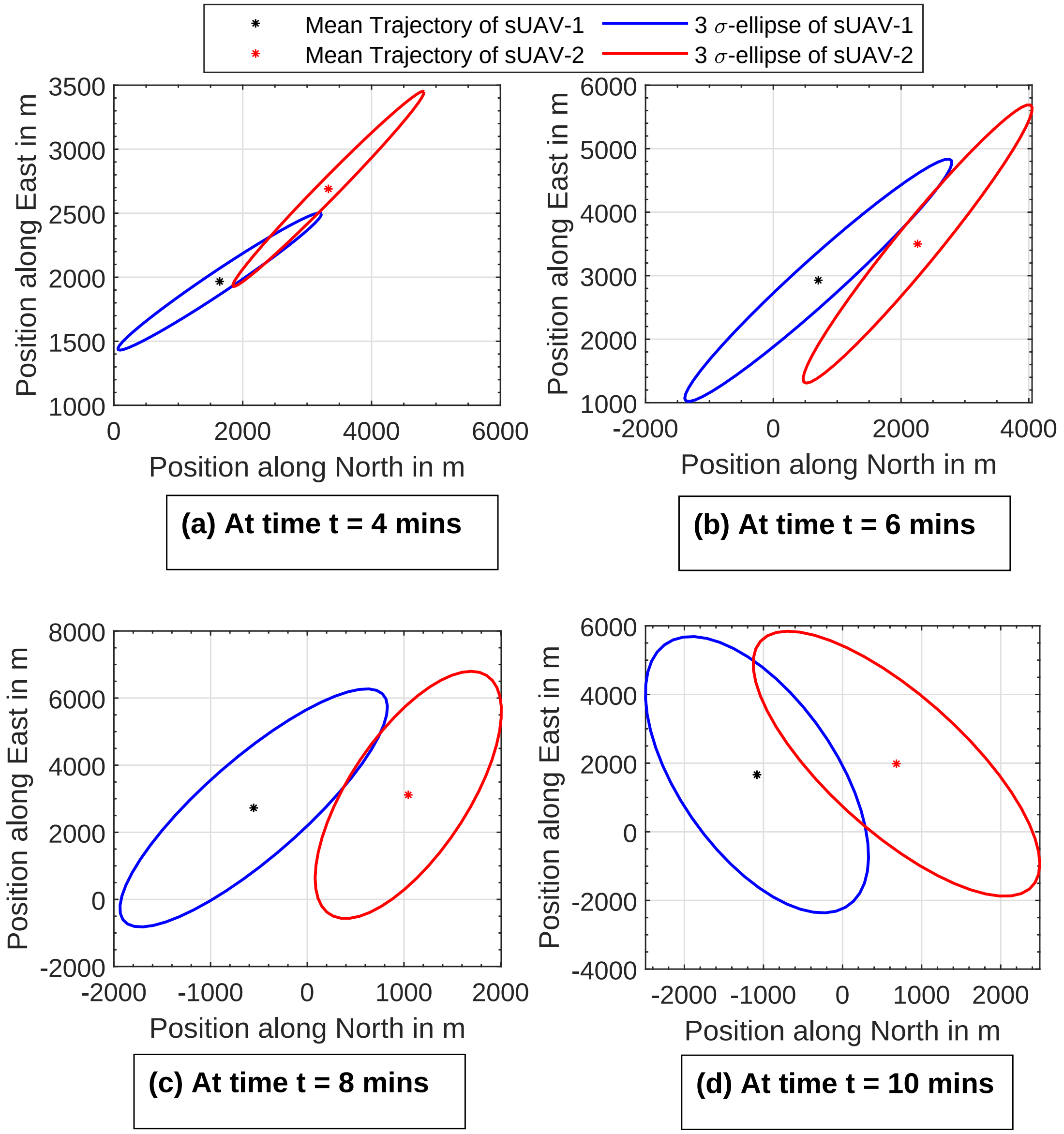


**Figure 8.—**Mean trajectory and 3-$\sigma$ confidence ellipses of sUAVs' positions (with larger uncertainty)

**Table 5.—**Commanded yaw acceleration profile for sUAVs

| sUAV index | Commanded yaw acceleration |
|---|---|
| sUAV-1 | $a_{yc} = \begin{cases} 0 & \text{if } t \leq 4 \text{ mins.} \\ -0.01g & \text{if } 4 \text{ mins.} < t < 5.5 \text{ mins.} \\ 0 & \text{else} \end{cases}$ |
| sUAV-2 | $a_{yc} = \begin{cases} 0 & \text{if } t \leq 4 \text{ mins.} \\ 0.01g & \text{if } 4 \text{ mins.} < t < 5 \text{ mins.} \\ 0 & \text{else} \end{cases}$ |
| sUAV-3 | $a_{yc} = \begin{cases} 0 & \text{if } t \leq 4 \text{ mins.} \\ -0.01g & \text{if } 4 \text{ mins.} < t < 5.15 \text{ mins.} \\ 0 & \text{else} \end{cases}$ |
| sUAV-4 | $a_{yc} = \begin{cases} 0 & \text{if } t \leq 3 \text{ mins.} \\ 0.01g & \text{if } 3 \text{ mins.} < t < 5.5 \text{ mins.} \\ 0 & \text{else} \end{cases}$ |



**(a) At time t = 4 mins**

**(b) At time t = 6 mins**

**(c) At time t = 8 mins**

**(d) At time t = 10 mins**

**Figure 9.—**Mean and 3-$\sigma$ confidence ellipses of sUAVs' positions at various time instants (with larger uncertainty)

gPC expansion technique to carry out uncertainty propagation.

### 4.2.2 Simulation Results

Figure 10 illustrates the temporal variation of the mean position and $3 - \sigma$ position confidence ellipses of the four sUAVs. Fig. 11 depicts the mean and $3 - \sigma$ confidence ellipses of the sUAVs' position at various time instants. Further, we also compute the bivariate probability density function of the trajectories of the sUAVs by generating large ensembles (=10,000) of the gPC solution. Fig. 12 illustrates the PDF contours of the position of the sUAVs along north and east. It should be noted that the PDF contours provide a more

**Table 6.**—Initial condition uncertainties

| Initial Position along | Distribution Type | PDF |
|---|---|---|
| North (sUAV-1) | Normal | $\mathcal{N}(0, 0.01^2)$ m |
| East (sUAV-1) | Normal | $\mathcal{N}(0, 0.01^2)$ m |
| North (sUAV-2) | Normal | $\mathcal{N}(0, 0.01^2)$ m |
| East (sUAV-2) | Normal | $\mathcal{N}(8000, 100^2)$ m |
| North (sUAV-3) | Normal | $\mathcal{N}(15000, 100^2)$ m |
| East (sUAV-3) | Normal | $\mathcal{N}(6500, 100^2)$ m |
| North (sUAV-4) | Normal | $\mathcal{N}(14000, 100^2)$ m |
| East (sUAV-4) | Normal | $\mathcal{N}(0, 0.01^2)$ m |

**Table 7.**—Parametric uncertainties

| Variables | Distribution Type | PDF | Mean value |
|---|---|---|---|
| $m$ | Uniform | $\mathcal{U}[3, 3.25]$ Kgs | 3.125 Kgs |
| $C_{D0}$ | Uniform | $\mathcal{U}[0.045, 0.05]$ | 0.047 |
| $S$ | Uniform | $\mathcal{U}[0.5, 0.525]$ m$^2$ | 0.5125 m$^2$ |
| $k$ | Uniform | $\mathcal{U}[0.0175, 0.0225]$ | 0.02 |
| $T_{\text{max}}$ | Uniform | $\mathcal{U}[3, 3.25]$ N | 3.125 N |

accurate representation of the probability of occurrence of the vehicle as compared to the $3-\sigma$ confidence ellipses. Clearly, from Figs. 10, 11, and 12, we can conclude that there is no risk of conflict between the sUAVs throughout the flight time.

To provide a quantitative assessment of the risk, we also calculate the probability of conflict between any two vehicles in the traffic by defining a separation tolerance $d_{\text{Tol}}$. To that end, the probability of conflict can be computed by calculating the cumulative distribution function (CDF) of the separation distance between any two vehicles (see the relation between CDF and probability measure in Section 2.1).We define the separation distance between any two vehicles $i$ and $j$, $i, j = 1, \dots, 4$, $i \neq j$ as follows

$$d_{ij}(t) = \sqrt{(X_i(t) - X_j(t))^2 + (Y_i(t) - Y_j(t))^2} \tag{42}$$

and calculate the probability, $\mathcal{P}\left(d_{ij}(t) \leq d_{\text{Tol}}\right) = \alpha$, $0 < \alpha < 1$.

Figure 13 illustrates the probability that the separation distance between any two sUAVs $i$ and $j$, $i, j = 1, \dots, 4$, $i \neq j$ is less than $d_{\text{Tol}} = 700$ m at any given instant of time. Clearly, the maximum probability that the separation distance between any two sUAVs is less than the tolerable limit of 700 m is $4.1\%$ (between sUAV-1 and sUAV-4).

### 4.2.3 Risk Analysis with larger uncertainties

From Section 4.1.3, we know that the separation distance between any 2 sUAVs is most influenced by uncertainties in maximum available thrusts, zero-lift drag coefficients, and wing planform area. Therefore, similar to the analysis carried out for the case with 2 sUAVs, we conduct the conflict risk assessment for 4 sUAVs with increased uncertainty bounds. Table 8 illustrates the distribution of uncertainties with increased uncertainty bounds in the variables with higher significance on safety metrics.

Figure 14 illustrates the temporal variation of the mean and $3-\sigma$ confidence ellipses of the sUAVs' position along north and east. Further, Fig. 15 depicts the mean and $3-\sigma$ confidence ellipses of the sUAVs'

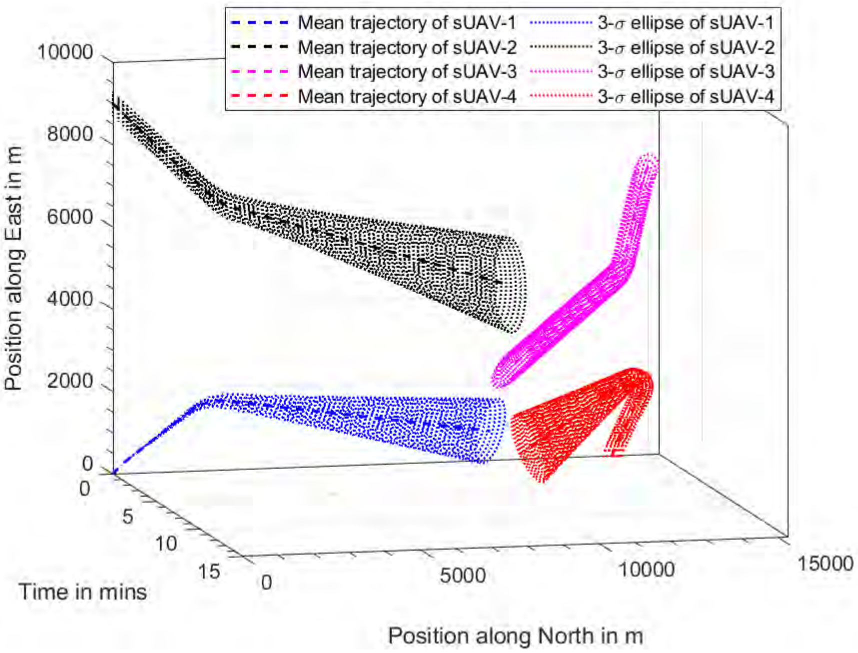


**Figure 10.—**Mean trajectory and $3-\sigma$ confidence ellipses of sUAVs' positions

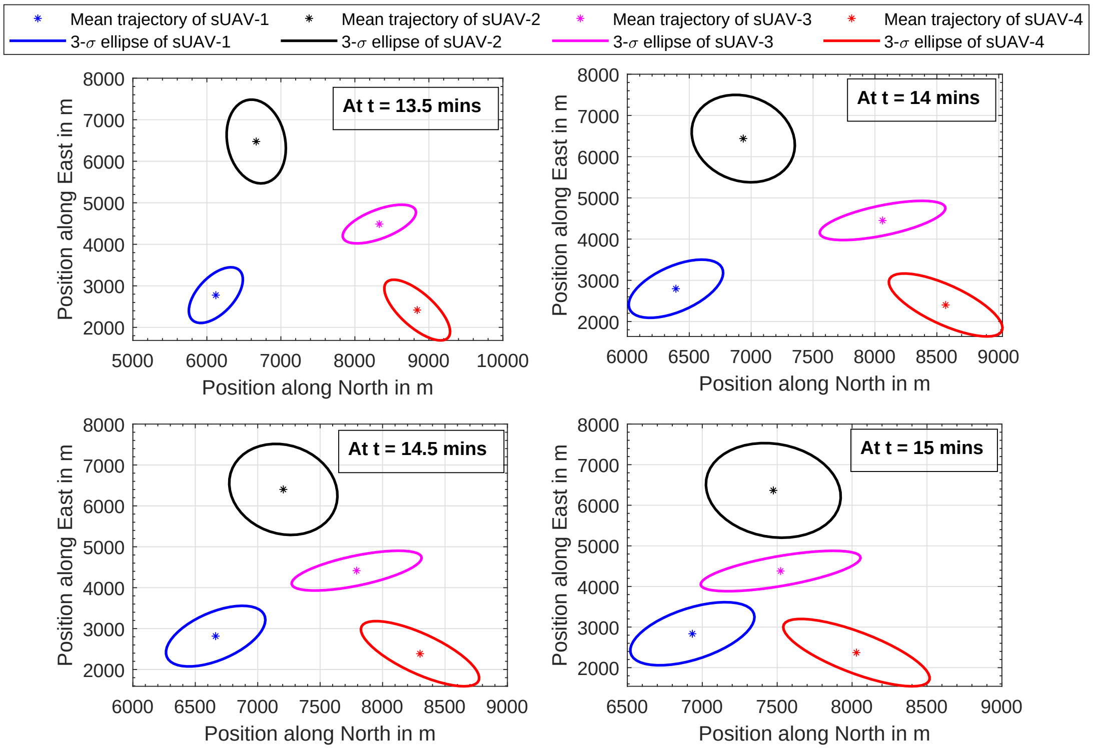


**Figure 11.—**Mean and $3-\sigma$ confidence ellipses of sUAVs' positions at various time instants

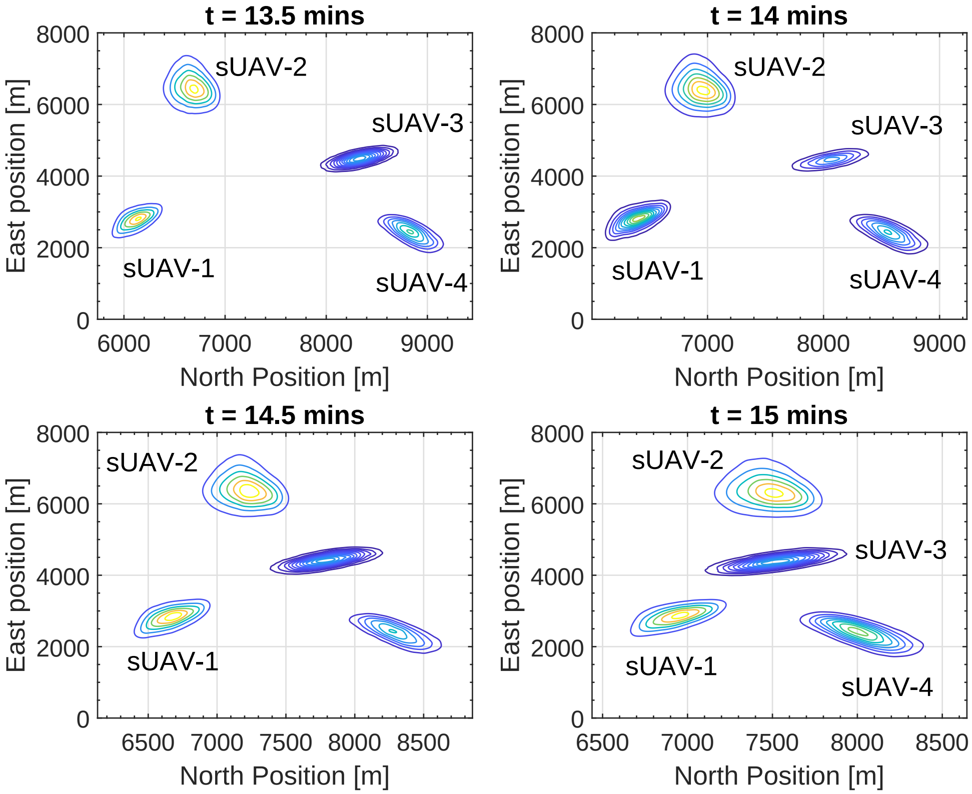


**Figure 12.—**PDF contours of sUAVs' positions at various time instants. The yellow contour lines correspond to the region with a high probability and the blue contour lines correspond to the region with a low probability.

**Table 8.—**Increased uncertainty bounds for significant uncertain variables

| Variables | Distribution Type | PDF | Mean value |
|---|---|---|---|
| $m$ | Uniform | $\mathcal{U}[3, 3.25]$ Kgs | 3.125 Kgs |
| $C_{D0}$ | Uniform | $\mathcal{U}[0.04, 0.055]$ | 0.047 |
| $S$ | Uniform | $\mathcal{U}[0.41, 0.615]$ m$^2$ | 0.5125 m$^2$ |
| $k$ | Uniform | $\mathcal{U}[0.0175, 0.0225]$ | 0.02 |
| $T_{\text{max}}$ | Uniform | $\mathcal{U}[2.75, 3.5]$ N | 3.125 N |

position along north and east at various time instants. Clearly, from Fig. 15, one can observe the colliding confidence ellipses in four sUAVs' positions, which depicts the increased risk of conflict among the traffic of sUAVs during the flight time. Further, Fig. 16 illustrates the PDF contours of the position of the sUAVs along north and east.

Figure 17 illustrates the probability that the separation distance (defined in Eq. (42)) between any two sUAVs $i$ and $j$, $i, j = 1, \ldots, 4$, $i \neq j$ is less than $d_{\text{Tol}} = 700$ m at any given instant of time. The maximum probability that the separation distance between any two sUAVs is less than the tolerable limit of 700 m is $25.6\%$ (between sUAV-1 and sUAV-4). Thus, the risk of conflict increases by approximately 6 folds when the uncertainties in the separation-sensitive parameters are increased from the PDFs defined in Table 7 to

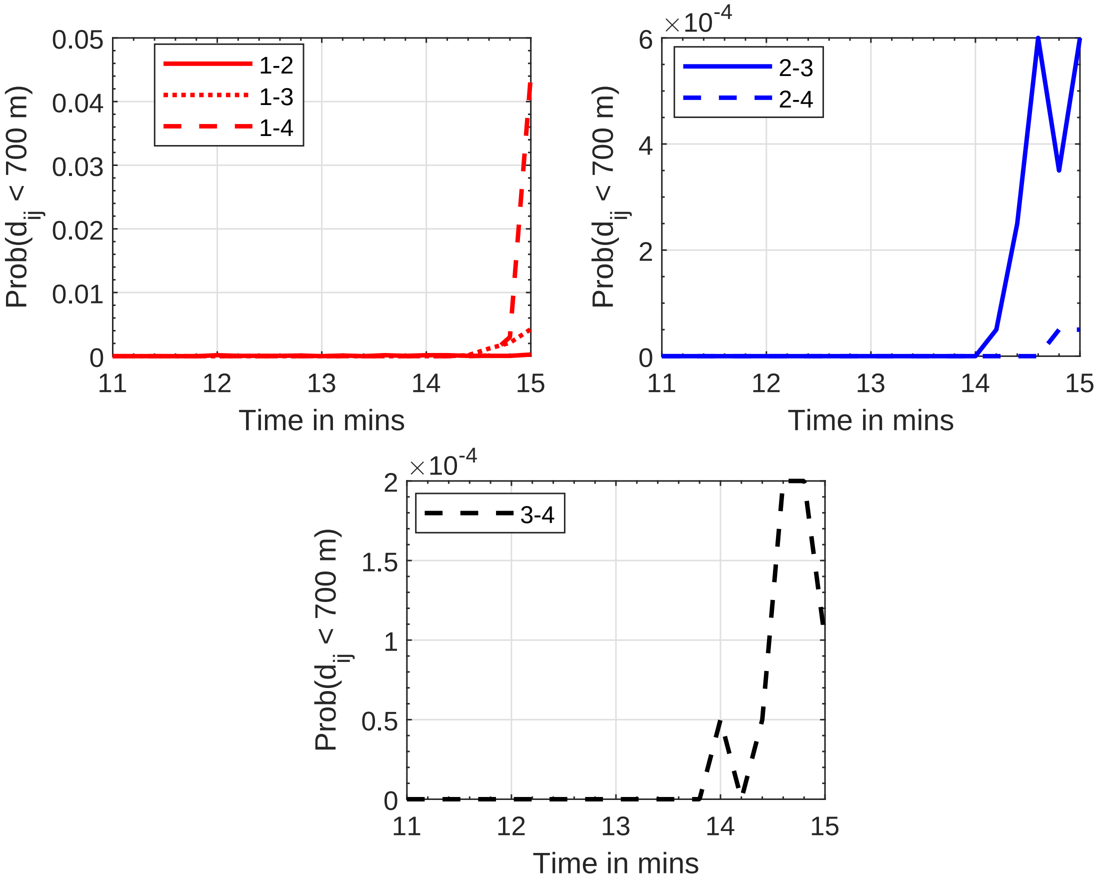


**Figure 13.—**Probability that the separation distance between any two sUAVs in the traffic is less than the tolerance of 700 m

the PDFs defined in Table 8.

# 5 Application to Conflict Assessment in an Airspace with Wind Field Uncertainties

This section applies the gPC expansion-based uncertainty propagation in a traffic of homogeneous as well as heterogeneous UAVs subject to wind uncertainties and assess the risk of conflict in the airspace.

## 5.1 Homogenous Airspace Models

We first consider an airspace with 4 identical fixed-wing UAVs namely, UAV-1, UAV-2, UAV-3, and UAV-4 operating in an uncertain wind field. All the four UAVs are assumed to demonstrate planar motions in the horizontal plane (North-East) at constant altitude as illustrated in Fig. 18.

**Simulation Parameters**

The four UAVs are assumed to have the following initial conditions:

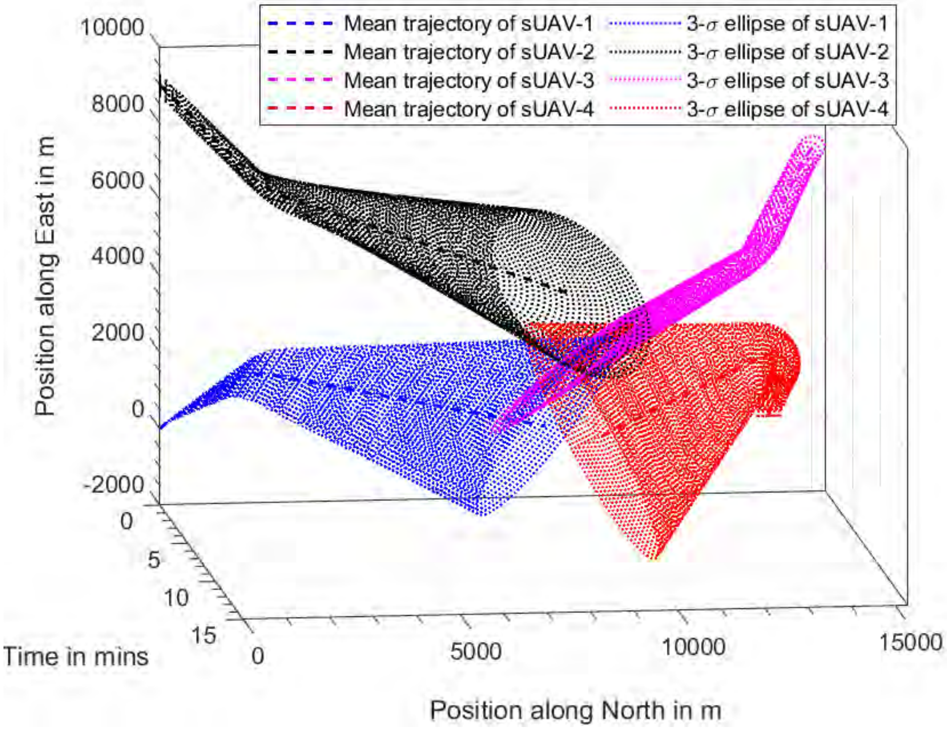


**Figure 14.—**Mean trajectory and $3-\sigma$ confidence ellipses of sUAVs' position (with increased uncertainty)

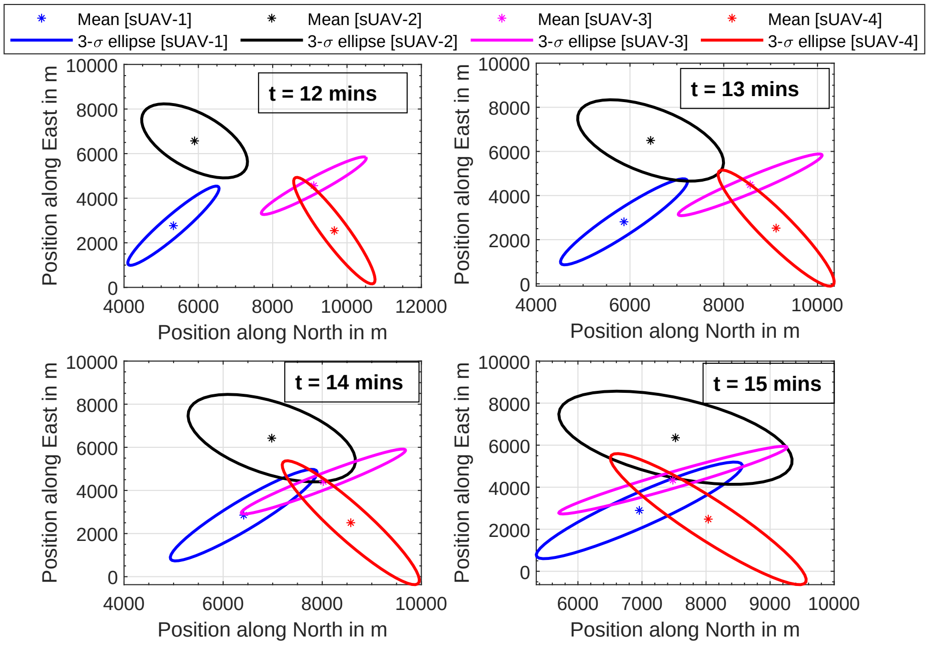


**Figure 15.—**Mean and $3-\sigma$ confidence ellipses of sUAVs' position at various time instants (with increased uncertainty)

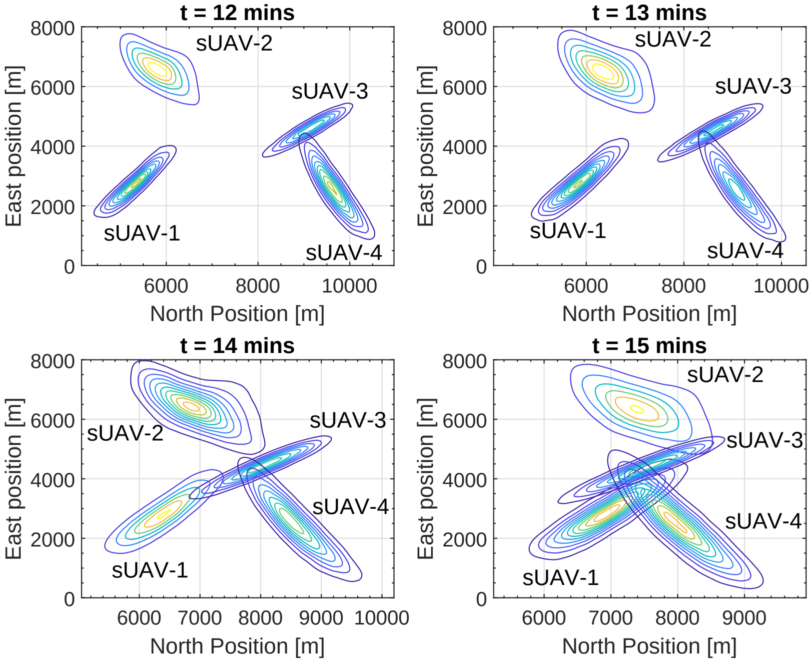


**Figure 16.**—PDF contours of sUAVs' positions at various time instants (with increased uncertainty). The yellow contour lines correspond to the region with a high probability and the blue contour lines correspond to the region with a low probability.

(i) for UAV-1, $x(0) = 0$ m, $y(0) = 0$ m, $V(0) = 13.4112$ m/s (30 mph), $\chi(0) = 0$ rad, $\eta(0) = 1$, and $\mu(0) = 0$ rad;

(ii) for UAV-2, $x(0) = 0$ m, $y(0) = 7000$ m, $V(0) = 13.4112$ m/s (30 mph), $\chi(0) = 0$ rad, $\eta(0) = 1$, and $\mu(0) = 0$ rad;

(iii) for UAV-3, $x(0) = 18000$ m, $y(0) = 7000$ m, $V(0) = 13.4112$ m/s (30 mph), $\chi(0) = \pi$ rad, $\eta(0) = 1$, and $\mu(0) = 0$ rad;

(iv) for UAV-4, $x(0) = 18000$ m, $y(0) = 0$ m, $V(0) = 13.4112$ m/s (30 mph), $\chi(0) = \pi$ rad, $\eta(0) = 1$, and $\mu(0) = 0$ rad.

Table 9 illustrates the parameters for the fixed wing UAVs and Table 10 gives the values of autopilot control gains. The velocity command for all the four UAVs is $V_c = 13.4112$ m/s (30 mph). For the four UAVs, the heading commands are as follows:

(i) For UAV-1 and UAV-3, $\chi_c = 0$ deg for $0 < t < 225$ s; it then increases at a constant rate of 0.9 deg per second (i.e., $\omega = 0.0157$ rad/s) for $225 < t < 270$ s. Finally it returns to constant heading for $270 < t < 600$ s.

(ii) For UAV-2 and UAV-4, $\chi_c = 0$ deg for $0 < t < 225$ s; it then decreases at a constant rate of 0.9 deg per second (i.e., $\omega = -0.0157$ rad/s) for $225 < t < 270$ s. Finally it returns to constant heading for $270 < t < 600$ s.

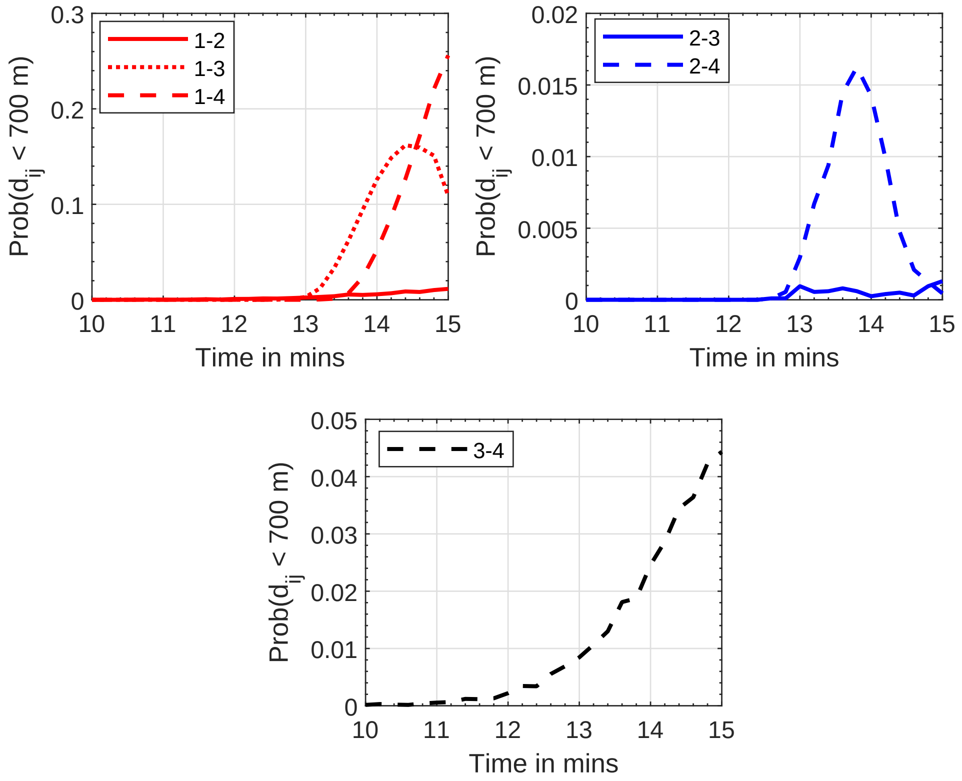


**Figure 17.**—Probability that the separation distance between any two sUAVs in the traffic is less than the tolerance of 700 m (with increased uncertainty)

**Table 9.**—Parameters of fixed wing UAVs

| Parameter | Notation | Value |
|---|---|---|
| Acc. due to gravity | $g$ | $9.81\ \mathrm{m/s^2}$ |
| Air density | $\rho$ | $1.225\ \mathrm{Kg/m^3}$ |
| Zero-lift drag coefficient | $C_{D_0}$ | 0.0475 |
| Wing area | $S$ | $0.5125\ \mathrm{m^2}$ |
| Drag polar constant | $K$ | 0.02 |
| Mass | $m$ | 3.125 Kg |
| Maximum thrust | $T_{\text{max}}$ | 5 N |
| Maximum lift coefficient | $C_{L_{\text{max}}}$ | 1.2 |
| Actuator time constant (throttle) | $\tau_\eta$ | 0.8 s |
| Actuator time constant (bank angle) | $\tau_\mu$ | 0.8 s |
| Structural limit on load factor | $n_{\text{max}}$ | 2 |

## 5.2 Uncertainty Propagation in UAVs' Trajectories due to Uncertainties in Wind Gradients

To predict the trajectories of the UAVs at any instant in the airspace accurately, it is essential to know the wind field or the spatial wind velocity gradients with certainty. However, only the probability distributions of wind velocity gradients are available for real-time applications. To assess the conflict risk among the

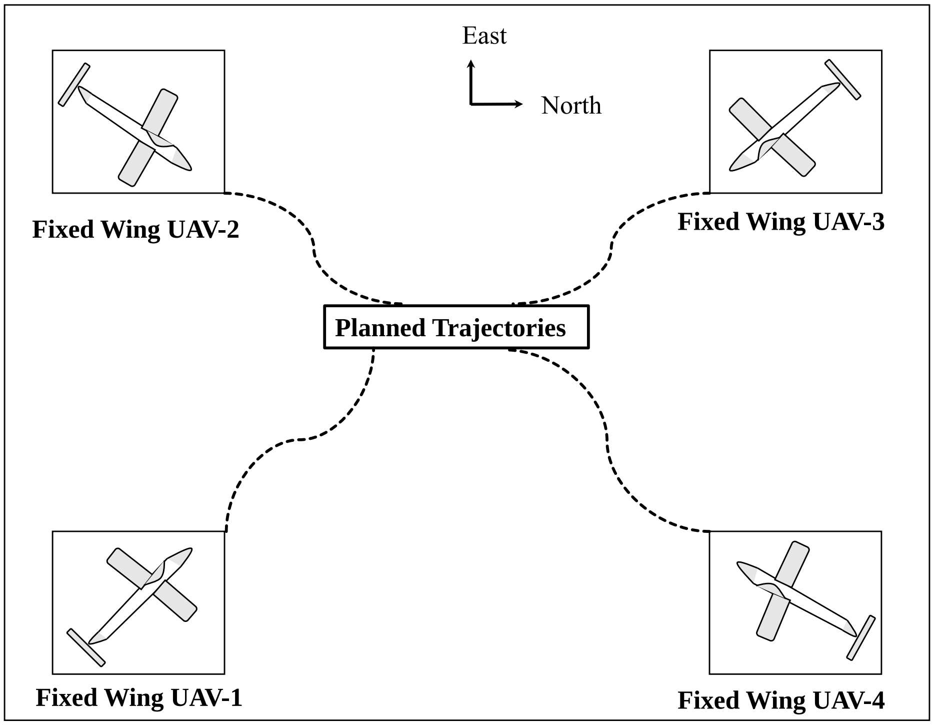


**Figure 18.—**Schematic of homogenous airspace

**Table 10.—**Autopilot control gains

| Controller | Notation and Value |
|---|---|
| Speed controller | $K_{V_P} = 1$, $K_{V_I} = 0.3$ |
| Heading controller | $K_{\mu_P} = 10$, $K_{\mu_I} = 5$ |

UAVs in the airspace, we carry out uncertainty propagation using gPC expansion framework.

Figure 19 illustrates the mean of the wind field assumed in this numerical simulation study and Table 11 lists the probability density functions of the uncertainties in the wind velocity gradients.

We first simulate the nominal trajectories of the UAVs by assuming that the wind field in the airspace is exactly known. To simulate the nominal trajectories (which gives the filed flight plan), we consider the wind field to be exactly the same as the mean wind field as illustrated in Fig. 19 which is obtained by considering deterministic values of spatial gradient equal to 0.001 for all four gradients. Figure 20 illustrates the nominal trajectories of the UAVs for a flight time of 10 minutes. Clearly, there is no conflict among the nominal trajectories of the UAVs in the airspace.

Next, we consider the scenario of a wind field, with uncertainties in the spatial gradients of the wind field. We consider third order $P = 3$ of gPC expansion with $N_q = 57$ sparse-grid collocation nodes to carry out uncertainty propagation in the trajectories of the UAVs. The response PDF contours of the trajectories of the UAVs at various time instants in depicted in Fig. 21. From Fig. 21, we can observe that the UAVs demonstrate conflict in the airspace after $t = 9$ minutes.

Further, to quantify the conflict between the UAVs, we calculate the probability of conflict between any two vehicles in the airspace. To that end, we define the minimum tolerable separation distance between any two vehicles as 800 m. From Fig. 22, we obtain that the risk of conflict in the airspace is maximum at

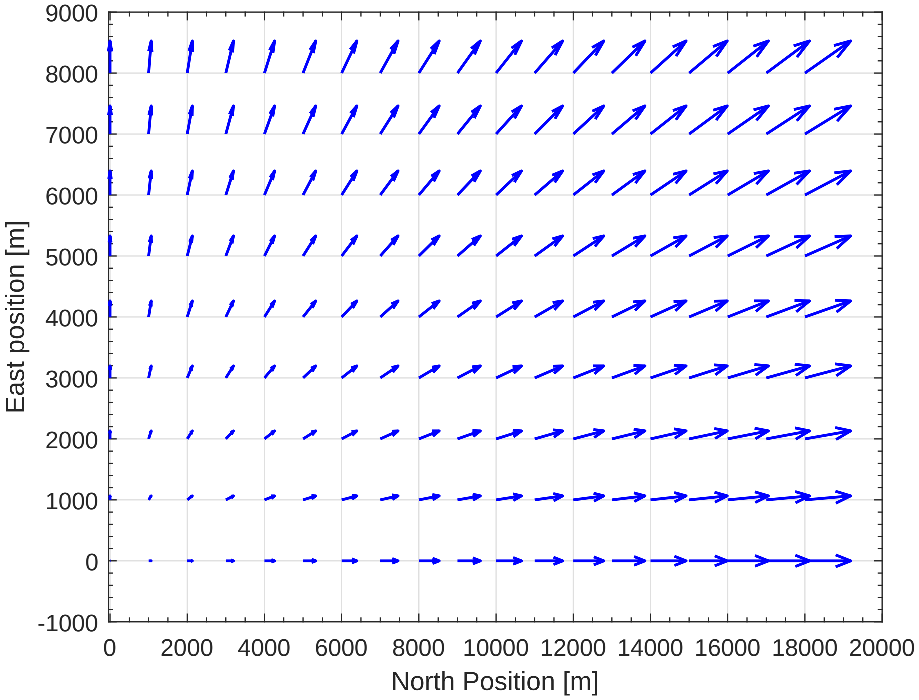


**Figure 19.**—Mean wind field obtained with the mean values of all 4 spatial gradients to be $0.001$. The blue arrows depict the magnitude and direction of the wind velocities.

**Table 11.**—Uncertainties in the wind velocity gradients

| Variables | Distribution Type | PDF |
|---|---|---|
| $\frac{\partial W_x}{\partial x}$ | Normal | $\mathcal{N}[0.001, 0.0001^2]$ s |
| $\frac{\partial W_x}{\partial y}$ | Normal | $\mathcal{N}[0.001, 0.0001^2]$ s |
| $\frac{\partial W_y}{\partial x}$ | Normal | $\mathcal{N}[0.001, 0.0001^2]$ s |
| $\frac{\partial W_y}{\partial y}$ | Normal | $\mathcal{N}[0.001, 0.0001^2]$ s |

$t = 9.5$ minutes.

$$\max\left(\mathcal{P}_{ij}\left(d_{ij}(t = 9.5 \text{ mins}) \leq d_{\text{Tol}}\right) = 0.0575 \tag{43}$$

The maximum probability that the separation distance between any two UAVs is less than the tolerable limit of 800 m is $5.75\%$ (between UAV-1 and UAV-2, and UAV-3 and UAV-4).

## 5.3 Heterogeneous Airspace Model

Next, we consider a heterogeneous airspace with 4 UAVs comprising of two fixed wing UAVs and two quadrotors in an uncertain wind field. We denote the vehicles as: Quadcopter-1, Quadcopter-2, Fixed Wing

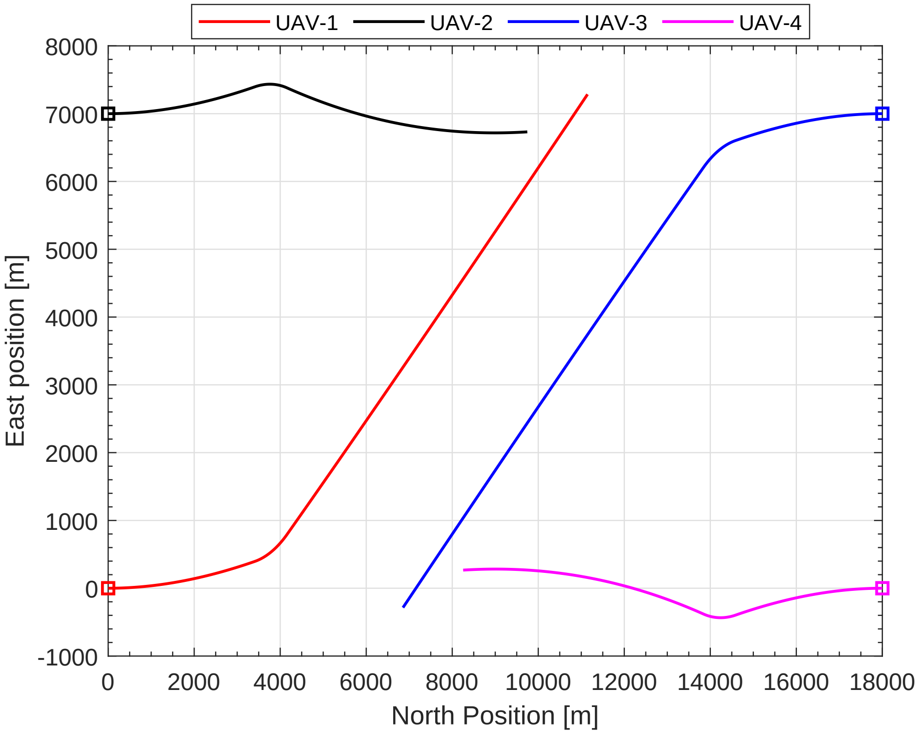


**Figure 20.—**Nominal trajectories of the UAVs. The colored squares are the initial locations of the UAVs.

UAV-1, and Fixed Wing UAV-2. The schematic of the airspace is illustrated in Fig. 23.

For the fixed-wing UAVs, the same equations of motion as in Eq. (31) are considered. The equations of motion of the quadrotor, their autopilots, and parameter configurations are described in Section 3.3

## 5.4 Simulation parameters

Both the fixed wing UAVs are assumed to have identical parameter configurations as provided in Table 9. In addition, both the quadrotors in the airspace are also assumed to be identical. The quadrotor parameters used for the simulation are provided in Table 12. The velocity command for all the four vehicles is $V_c = 13.4112$ m/s (30 mph). The heading commands for Quadrotor-1 and Quadrotor-2 are similar to that of UAV-1 and UAV-3 while that for Fixed Wing UAV-1 and Fixed Wing UAV-2 are similar to that of UAV-2 and UAV-4 (see the case of homogeneous airspace), respectively. Further, the control gains of the speed, heading, and attitude controllers in the quadrotor autopilots are provided in Table 13. The four vehicles are assumed to have the following initial conditions:

(i) for Quadrotor-1, $x(0) = 0$ m, $y(0) = 0$ m, $z(0) = 20$ m, $\dot{x}(0) = 13.4112$ m/s (30 mph), $\dot{y}(0) = 0$ m/s , $\dot{z}(0) = 0$ m, $\phi(0) = \theta(0) = \psi(0) = 0$ rad, $\dot{\phi}(0) = \dot{\theta}(0) = \dot{\psi}(0) = 0$ rad/s;

(ii) for Fixed Wing UAV-1, $x(0) = 0$ m, $y(0) = 7000$ m, $V(0) = 13.4112$ m/s (30 mph), $\chi(0) = 0$ rad, $\eta(0) = 1$, and $\mu(0) = 0$ rad;

(iii) for Quadrotor-1, $x(0) = 18000$ m, $y(0) = 7000$ m, $z(0) = 20$ m $\dot{x}(0) = -13.4112$ m/s (-30 mph), $\dot{y}(0) = 0$ m/s , $\dot{z}(0) = 0$ m/s, $\phi(0) = \theta(0) = \psi(0) = 0$ rad, $\dot{\phi}(0) = \dot{\theta}(0) = \dot{\psi}(0) = 0$ rad/s;

(iv) for Fixed Wing UAV-2, $x(0) = 18000$ m, $y(0) = 0$ m, $V(0) = 13.4112$ m/s (30 mph), $\chi(0) = \pi$ rad, $\eta(0) = 1$, and $\mu(0) = 0$ rad.

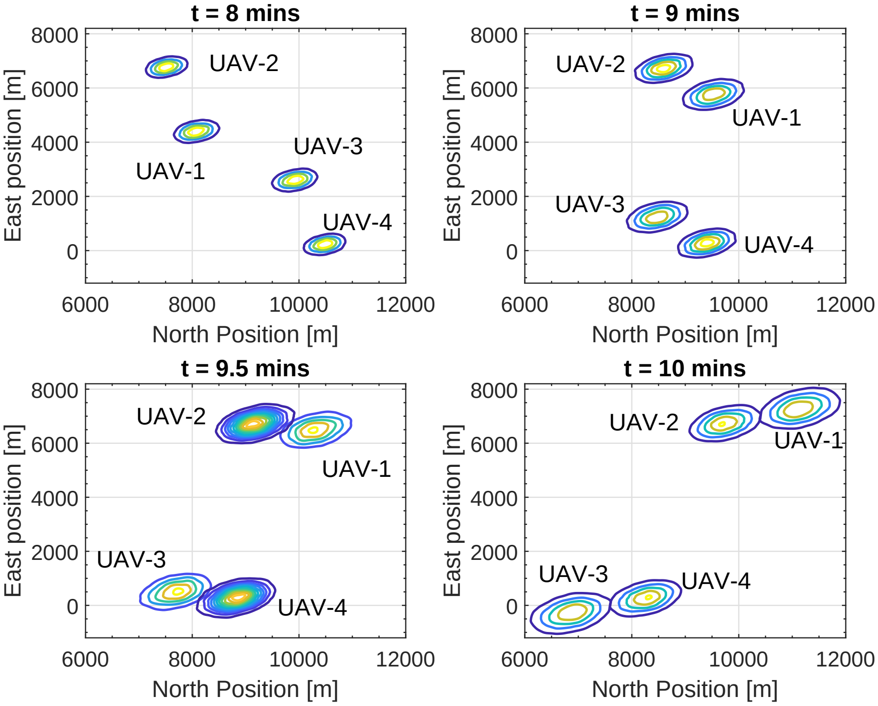


**Figure 21.—**PDF contours of the positions of the fixed-wing UAVs at various time instants. The yellow and blue regions depict the high and low probability regions of UAV's location, respectively

**Table 12.—**Parameters of Quadrotors

| Parameter | Nominal Value |
|---|---|
| $g$ | 9.81 $\mathrm{m/s^2}$ |
| $m$ | 0.4794 Kg |
| $I_x$ | 0.0086 Kg $\mathrm{m}^2$ |
| $I_y$ | 0.0086 Kg $\mathrm{m}^2$ |
| $I_z$ | 0.0172 Kg $\mathrm{m}^2$ |
| $J_r$ | $3.7404 \times 10^{-5}$ Kg $\mathrm{m}^2$ |
| $b$ | $3.13 \times 10^{-5}$ |
| $d$ | $9 \times 10^{-7}$ |
| $L$ | 0.225 m |

## 5.5 Uncertainty Propagation in the Heterogeneous Airspace due to Uncertainties in Wind Gradients

For the airspace containing two fixed wing UAVs and two quadrotors, we consider the uncertainties in the spatial gradients of the wind field. The distributions of the wind gradients are assumed to be the same as in the earlier case with homogeneous airspace (see Table 11 and Fig. 19).

We consider third order $P = 3$ of gPC expansion with $N_q = 57$ sparse-grid collocation nodes to carry out uncertainty propagation in the trajectories of the fixed wing UAVs and quadrotors. The response PDF contours of the trajectories of the vehicles at various time instants is depicted in Fig. 24. Clearly, we can

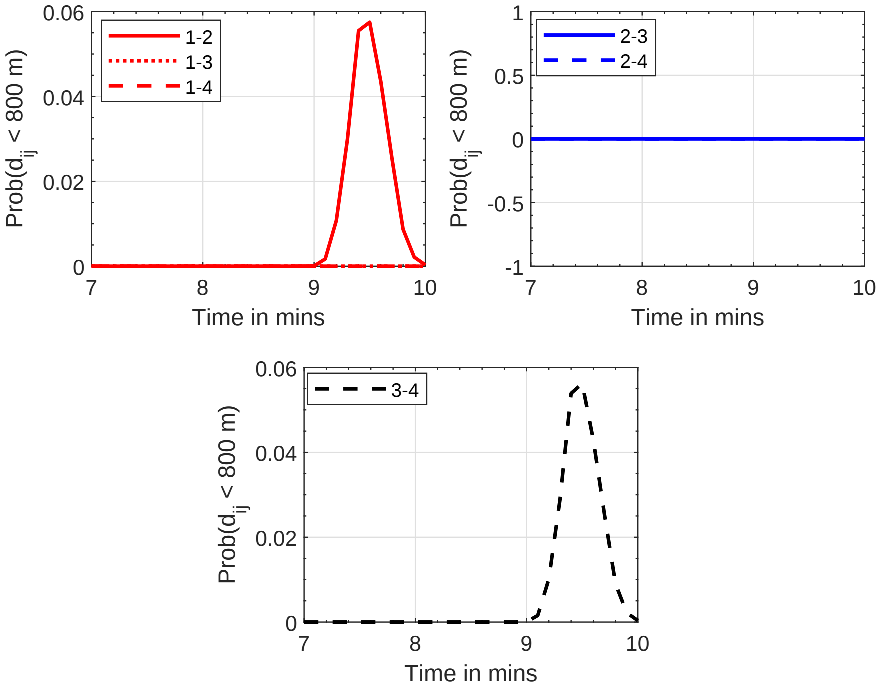


**Figure 22.—**Probability that the separation between any two UAVs in the airspace is less than the tolerance of 800 m

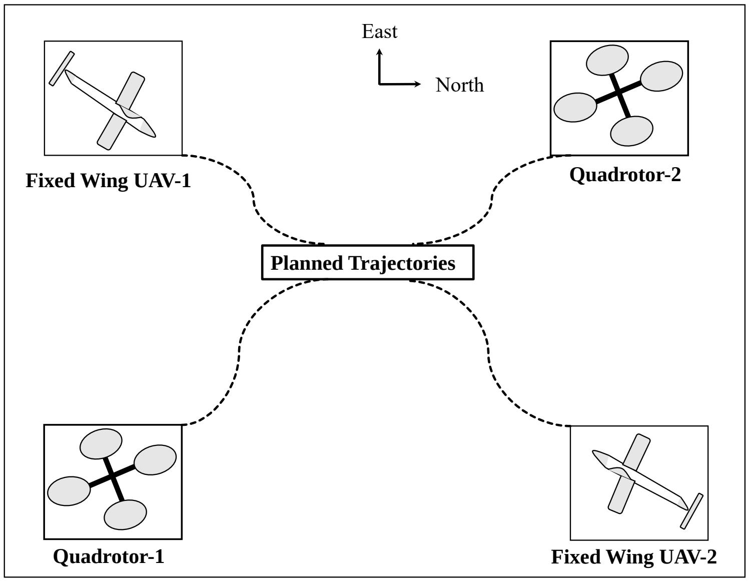


**Figure 23.—**Schematic of heterogeneous airspace

observe probable conflict in the trajectories of the vehicles in the airspace after $t = 10$ minutes.

We further quantify the risk of conflict in the airspace by calculating the probability that the separation

**Table 13.—**Autopilot control gains

| Notation | Value |
|---|---|
| $K_{\phi_P}$=$K_{\theta_P}$= $K_{\psi_P}$ | 10 |
| $K_{\phi_D} = K_{\theta_D} = K_{\psi_D}$ | 5 |
| $K_{v_x}$ | 12 |
| $K_{v_y} = K_{v_z}$ | 10 |

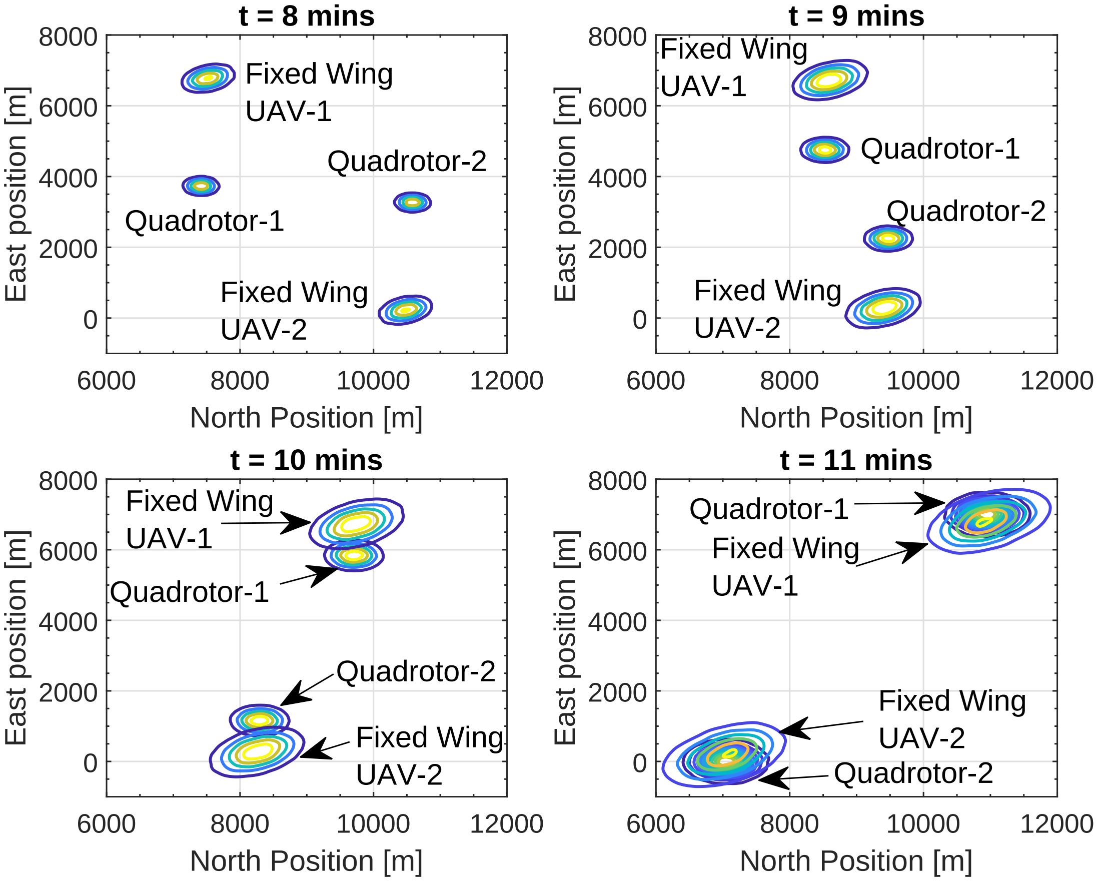


**Figure 24.—**PDF contours of the positions of the quadrotors and fixed-wing UAVs at various time instants. The yellow and blue regions depict the high and low probability regions of UAV's location, respectively

distance between any two vehicles in the airspace is less than a safe separation distance. Figure 25 illustrates the temporal variation of probability of conflict by considering $d_{\text{Tol}} = 800$ m. The maximum probability that the separation distance between any two vehicles is less than the tolerable limit of 800 m is 98.93 % (between Quadrotor-1 and Fixed Wing UAV-1, and Quadrotor-2 and Fixed Wing UAV-2).

In comparison with the results from the homogenous case of 4 fixed wing UAVs we observe that there is a significantly increased probability that the minimum separation distance will be breached by UAVs of different types operating in the same environment. Note, the winds are same (in terms of uncertainties), but the manner in which the winds affect the heading/course of the vehicles is different. The heading/course of

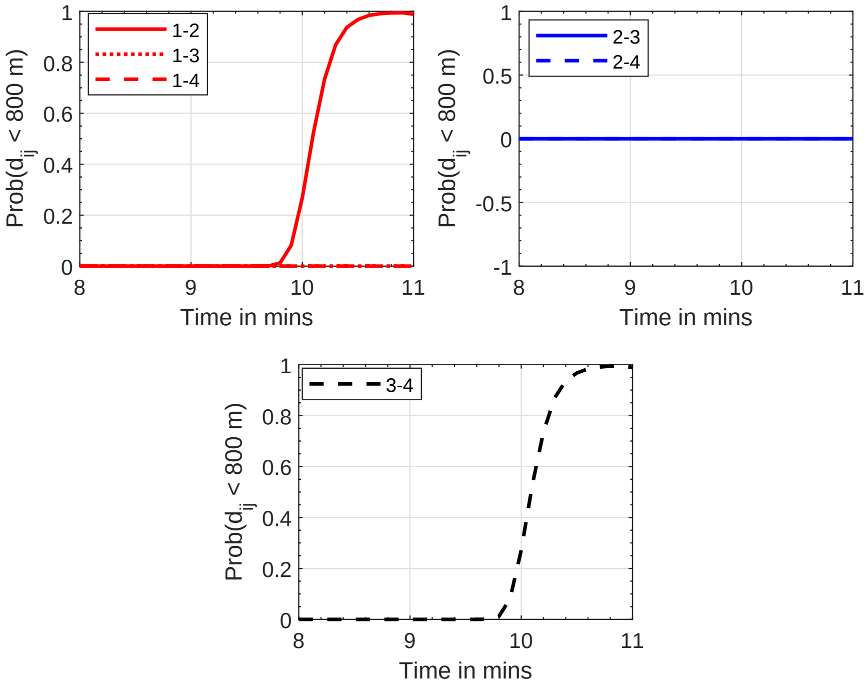


**Figure 25.—**Probability that the separation between any two vehicles is less than the tolerance of 800 m

the fixed wing UAVs are directly affected by the wind components, while the quadrotor only sees these from the control torques (compensatory coupling).

## 5.6 Temporal Assessment of Computational Cost

The UQ framework is developed in a distributed traffic risk assessment setting. Therefore, for all the analysis carried out in this report, the time complexity of the computation should be linear, i.e. the order of computational complexity of $\mathcal{O}(N)$ where $N$ is the number of vehicles in the traffic. To verify this, we assess the temporal computational cost for different problem cases pertinent to characterizing the risk of conflict in the airspace. The results listed below are produced from MATLAB R2021a on an Intel Core i7 10th Gen CPU @230 GHz.

### Case 1 - Traffic of open loop fixed wing sUAVs with uncertain parameters and initial positions

For this case, we have $9$ states and $7$ uncertain variables. For, the fourth-order gPC expansion with $127$ collocation nodes, the computational time for uncertainty propagation in a single vehicle is $3.56$ seconds. Table 14 lists the computation time to compute conflict risk among different numbers of vehicles in the traffic.

### Case 2 - Traffic of fixed wing sUAVs with autopilot in uncertain wind field (no other uncertainties)

For this case, we have $11$ states and $4$ uncertain variables. For, second-order gPC expansion with $9$ collocation nodes, the computation time for uncertainty propagation in a single vehicle is $4.23$ seconds. Table 15 lists the computation time to compute conflict risk among different numbers of vehicles in the

**Table 14.—**Computation Time

| Number of vehicles | Computation time for a horizon of 10 minutes |
|---|---|
| 2 | 7.14 seconds |
| 3 | 10.89 seconds |
| 4 | 11.16 seconds |

traffic.

**Table 15.—**Computation time

| Number of vehicles | Computation time for a horizon of 10 minutes |
|---|---|
| 2 | 8.56 seconds |
| 3 | 12.98 seconds |
| 4 | 17.08 seconds |

**Case 3 - Traffic of quadrotors with autopilot in uncertain wind field (no other uncertainties)**

For this problem, we have $14$ states and $4$ uncertain variables. For, second-order gPC expansion with 9 collocation nodes, the computation time for uncertainty propagation in a single vehicle is $21.90$ seconds. Table 16 lists the computation time to compute conflict risk among different numbers of vehicles in the traffic.

**Table 16.—**Computation time

| Number of vehicles | Computation time for a horizon of 10 minutes |
|---|---|
| 2 | 43.92 seconds |
| 3 | 65.78 seconds |
| 4 | 87.95 seconds |

It can be seen that the computation time for Cases 2 and 3 are almost linear. These cases have fewer uncertain inputs (four) and need fewer collocation nodes for UQ. This is different for Case 1, where the number of uncertain inputs and the collocation nodes are higher.

# 6 Estimation of the Probability of Conflict (Confidence Region) for Multiple sUAVs

In this section, we seek to dynamically compute confidence regions for sUAV positions. This is done within the gPC based UQ framework by using an Ensemble Filter (EnKF) (Ref. 38, 39) to update the sUAVs' positions within the traffic. We first compute the position covariance contours for each sUAVs using the propagation step in the Ensemble filter, followed by a one-time measurement update (GPS) for the ensemble to show how the position covariance contours change. The overlap/intersection of these covariance "ellipses" indicate potential conflict. Within the scope of this work, this computation is helpful since it allows us to compute future potential conflicts in the airspace for example, situations where there is loss of navigation precision or measurement updates (GPS degradation).

As noted in Ref. 38, 39, the correction step of the gPC-based EnKF filter only involves algebraic manipulations and low computational burden. Besides, the computational complexity of the filter depends upon the number of collocation nodes used to compute the coefficients of the gPC expansion. Thus, for a given order of gPC expansion, the higher the efficacy of the quadrature rule, the lower would be the computational cost of the filtering algorithm. The filter from Ref. 38, 39 is applied to an air traffic scenario involving multiple sUAVs.

### Application to Confidence Region Determination

In this scenario, we consider the traffic of four open-loop fixed-wing sUAVs demonstrating wing-level turning maneuvers at constant altitudes. The sUAVs are assumed to follow the dynamics as given in Section 3.1 with the commanded yaw acceleration values as provided in Table 5. The initial condition uncertainties and parametric uncertainties in the sUAVs are as listed in Table 6 and Table 7, respectively.

To compute the filter estimates, we consider simplified GPS position measurements of the sUAVs which are modeled as follows:

$$\begin{bmatrix} \tilde{X} \\ \tilde{Y} \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} + \boldsymbol{\eta} \tag{44}$$

where $\boldsymbol{\eta}\, in\, \mathbb{R}^2$ is the measurement noise (zero-mean Gaussian with specified covariance). The covariance matrix of the measurement error is given by

$$\mathbf{R} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \mathrm{m}^2 \tag{45}$$

To illustrate the methodology, we assume that a GPS measurement is available at $t = 5$ mins only. We propagate the uncertainties in the position until the measurements are available and update the position uncertainty once the GPS measurements are available.

In order to understand the effect of the measurement update on the position uncertainty, we first estimate the posterior PDF of sUAV-3 only. When the measurements are obtained, the gPC-based EnKF is utilized to estimate the position PDF at $t = 5$ minutes. At other instants of time, the gPC-based propagation step is followed. The resultant temporal propagation of the posterior PDF of the sUAV-3's trajectory is depicted in Figure 26. Clearly, at $t = 5$ minutes, the position uncertainty is reduced significantly.

Next, we compute the confidence region of all four sUAVs at various time instants. The obtained posterior confidence regions for the sUAVs are illustrated in Figure 27. It can be seen that, although the one-time GPS-based measurement update reduces the position uncertainty, the uncertainty grows enough after the correction step for the sUAVs-1, 3, and 4 and demonstrates the possibility of conflict.

Further, we compute the probability of conflict between a pair of sUAVs for the case with a one-time measurement update and compare it with the case where the measurement is not available. Figure 28 depicts the probability of conflict (with a distance tolerance of 700 m) for the case where the GPS measurement was not available. It can be verified that the maximum probability of conflict is around 26% (between vehicles 1 and 4). Fig. 29 depicts the probability of conflict for the case where the GPS measurements were available at $t = 5$ minutes. For the case where GPS measurements were available, the maximum probability of conflict is around 7.5% (between vehicles 3 and 4). Clearly, the overall probability of conflict is reduced when the measurement-based update of the predicted confidence region is carried out at $t = 5$ minutes.

In summary, with more frequent measurement updates, the probability of conflict reduces and can be predicted for all the vehicles in the air traffic. It is also possible to simulate potential conflicts for varying measurement update rates. This is particularly helpful when multiple vehicles move around in an urban

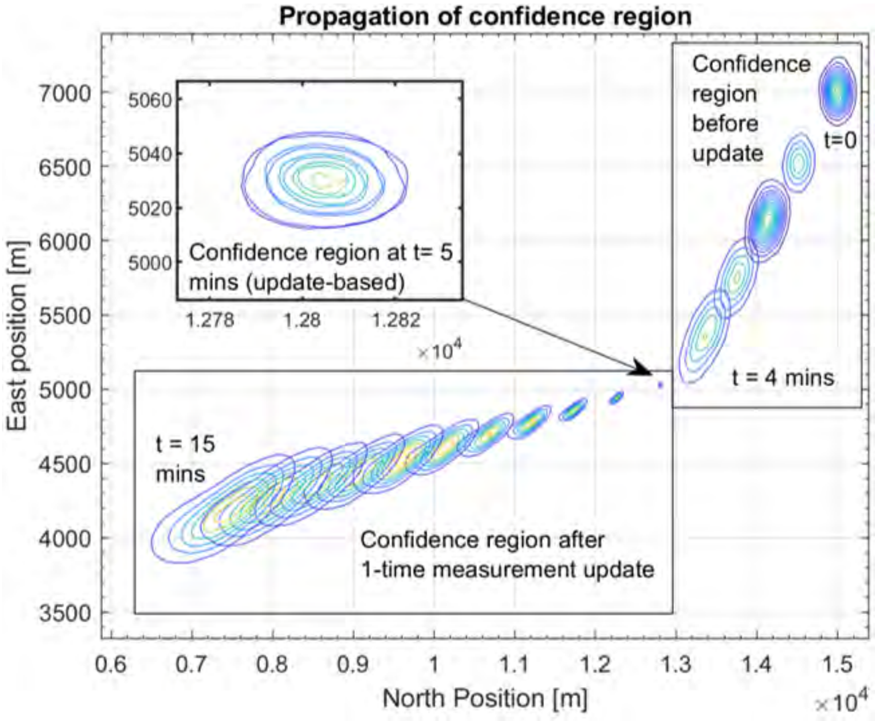


**Figure 26.—**Confidence region of sUAV-3's position throughout flight time

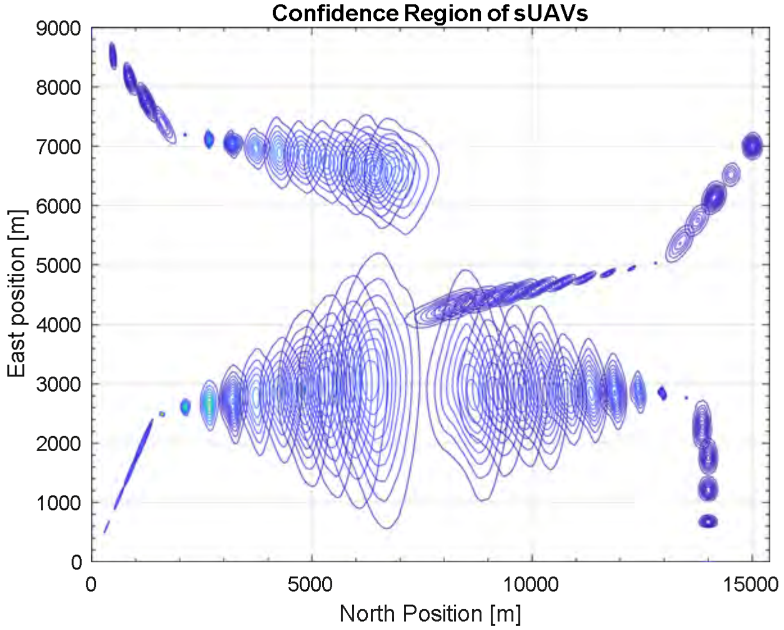


**Figure 27.—**Confidence region of all four sUAV's position throughout flight time

cluttered environment wherein GPS is typically degraded. It allows for a proactive separation assurance determination for the vehicles in operation.

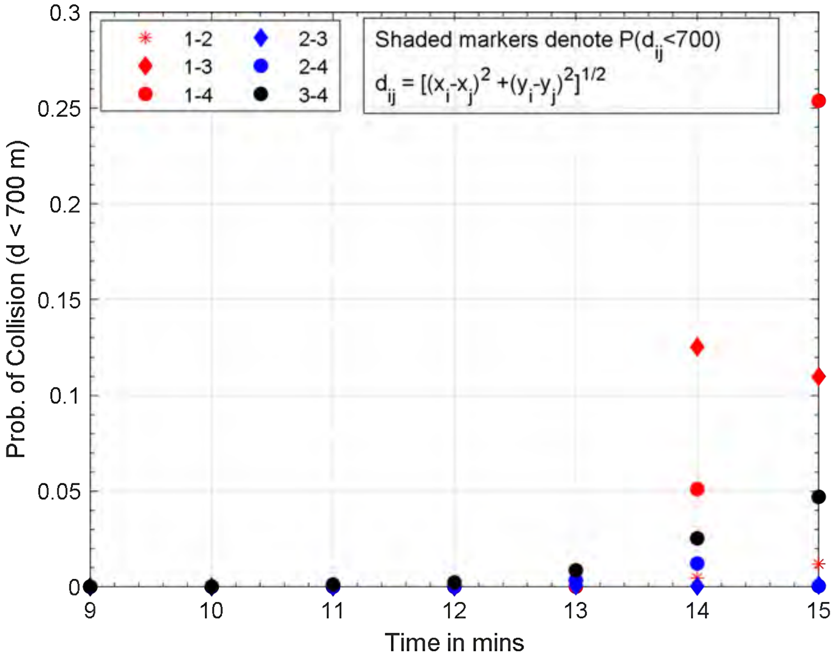


**Figure 28.—**Probability of conflict among sUAVs without measurement correction

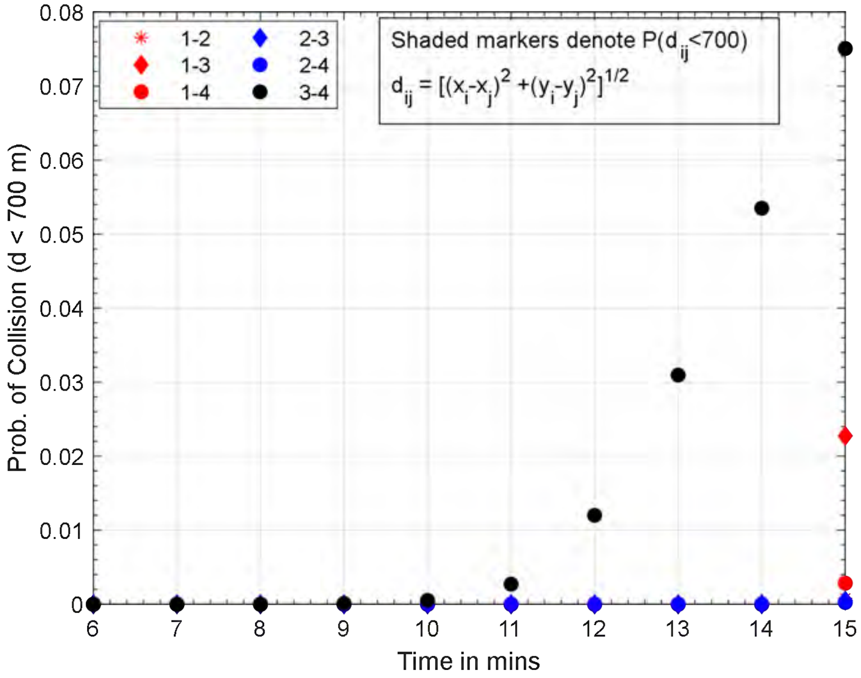


**Figure 29.—**Probability of conflict among sUAVs with one-time measurement correction at $t = 5$ minutes

# 7 Maximum Uncertainty Distribution in a Traffic of Small Unmanned Aerial Vehicles For Conflict-free Airspace Operation

In this section, we develop the framework further to compute the maximum allowable uncertainty knowing the position covariance information from multiple sUAVs. Note, the computation of the position covariance "ellipses" was shown in the previous section.

## 7.1 Problem Description

We consider a filed flight plan with wings level, constant-altitude motion for all the sUAVs. Also, the flight plan for any $i$th sUAV is generated with a constant commanded yaw acceleration.

### Confidence ellipses and risk of conflict

In this work, we consider uncertainties in the parameters of the system model of sUAVs described Section 3. *We assume that all the vehicles have sets of parameters which are uncertain and have identical probability density functions for all the vehicles,* $i = 1, 2, \ldots, M$. Due to uncertainties in the system, the governing equations of motion becomes stochastic in nature. Therefore, a distribution of the $X_i - Y_i$ for all $i = 1, 2, \ldots, M$ trajectories (along North and East coordinates, respectively) are obtained at the constant altitudes of the sUAVs. Further, one can also compute the statistics of the trajectories of the sUAVs. In this section, we denote the mean trajectory vector of $i$th sUAV as $\boldsymbol{\mu}_{XY_i}(t) = [\mu_{X_i}(t), \mu_{Y_i}(t)]^{\mathrm{T}}$ where

$$\mu_{X_i}(t) = \mathbb{E}[X_i(t)], \quad \mu_{Y_i}(t) = \mathbb{E}[Y_i(t)].$$

We define the confidence ellipse of the position of the $i$th sUAV at time instant $t$ as a set given as follows

$$E_i(t) := \left\{ \mathbf{x} : (\mathbf{x} - \boldsymbol{\mu}_{XY_i}(t))^{\mathrm{T}} \mathbf{P}_i^{-1}(t)(\mathbf{x} - \boldsymbol{\mu}_{XY_i}(t)) \right\} = 1 \tag{46}$$

where $\mathbf{P}_i(t) = \mathbb{E}[\mathbf{X}_i(t)\mathbf{Y}_i(t)^{\mathrm{T}}] - \boldsymbol{\mu}_{XY_i}(t)\boldsymbol{\mu}_{XY_i}(t)^{\mathrm{T}}$ is the covariance matrix. The $3-\sigma$ confidence ellipse is obtained by scaling the major and minor axis (i.e. along the eigenvector of $\mathbf{P}_i$) of the confidence ellipse in Eq. (46) by 3 folds. We denote the $3-\sigma$ confidence ellipse of the $i$th sUAV at time instant $t$ as $E_{3\sigma_i}(t)$.

In this context, one can propagate the uncertainties in the system using a suitable uncertainty propagation technique (such as Monte Carlo-based repetitive simulation and gPC expansion technique) and obtain mean trajectories and $3-\sigma$ confidence ellipses for the $X_i - Y_i$ position for all $i = 1, 2, \ldots, M$ sUAVs in the traffic.

Let $T$ be the time horizon for the sUAVs under consideration. Then the *objective of the problem* is to compute the maximum distribution of the parametric uncertainties in the system such that the $3-\sigma$ confidence ellipses of the trajectories, $E_{3\sigma_i}(t)$ and $E_{3\sigma_j}(t)$ of any two sUAVS $i$ and $j$, for all $i, j = 1, 2, \ldots, M, i \neq j$ do not intersect with each other at any time instant $t \in [0, T]$.

In this work, we define such a scenario as a "no-conflict condition". As stated earlier, we restrict the probability density functions (PDFs) of the uncertainties to being Gaussian and Uniform. Further, we define the maximum distribution as follows: (i) maximum standard deviation for normally distributed parameters and (ii) maximum difference between upper and lower bounds for uniformly distributed parameters. Figs. 30 and 31 exemplify the $3-\sigma$ confidence ellipses of two sUAVs at a time instant $t$ that respectively satisfy and violate the no-conflict condition.

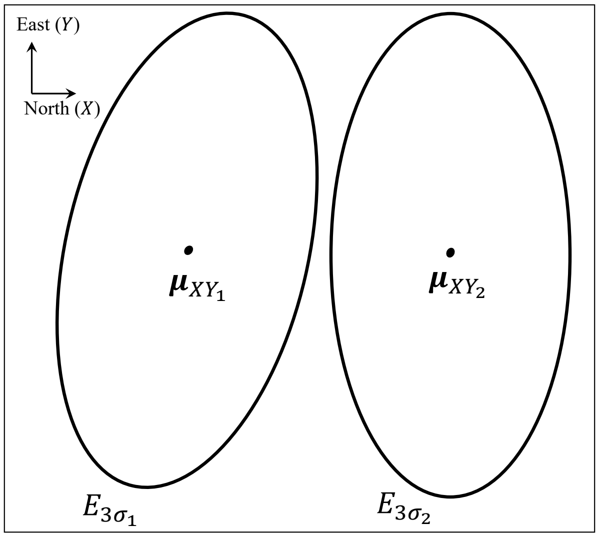


**Figure 30.—**$3-\sigma$ confidence ellipses of two sUAVs satisfying no-conflict condition

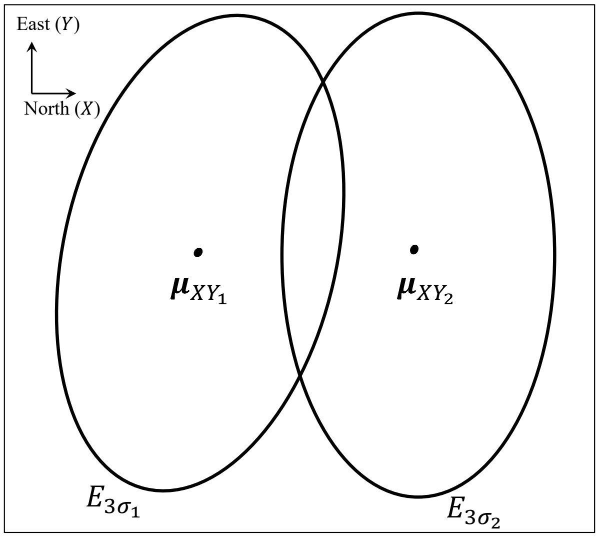


**Figure 31.—**Intersecting $3-\sigma$ confidence ellipses of two sUAVs depicting risk of conflict

## 7.2 No-conflict condition in sUAVs: An Ellipsoidal Intersection Approach

In this section, we first discuss conditions for the intersection of the two ellipsoids in $n$-dimensions and then propose our results to obtain the maximum distribution of parametric uncertainties for the no-conflict condition.

### 7.2.1 Ellipsoidal Intersection

Let us consider two ellipsoids $E_A(\mathbf{a},\mathbf{A})$ and $E_B(\mathbf{b},\mathbf{B})$ which can be defined as following

$$E_A(\mathbf{a},\mathbf{A}) := \Big\{\mathbf{x} : (\mathbf{x}-\mathbf{a})^{\mathrm{T}}\mathbf{A}(\mathbf{x}-\mathbf{a}) \leq 1\Big\}, \quad E_B(\mathbf{b},\mathbf{B}) := \Big\{\mathbf{x} : (\mathbf{x}-\mathbf{b})^{\mathrm{T}}\mathbf{A}(\mathbf{x}-\mathbf{b}) \leq 1\Big\} \tag{47}$$

where $\mathbf{a} = [\mu_{1_a}, \ldots, \mu_{n_a}]^{\mathrm{T}} \in \mathbb{R}^n$ and $\mathbf{b} = [\mu_{1_b}, \ldots, \mu_{n_b}] \in \mathbb{R}^n$ and $\mathbf{B} > 0 \in \mathbb{R}^n$ are the centers and $\mathbf{A} > 0 \in \mathbb{R}^n$ and $\mathbf{B} > 0 \in \mathbb{R}^n$ are the shape matrices of the $n$-dimensional ellipsoids $E_A(\mathbf{a}, \mathbf{A})$ and $E_B(\mathbf{b}, \mathbf{B})$, respectively.

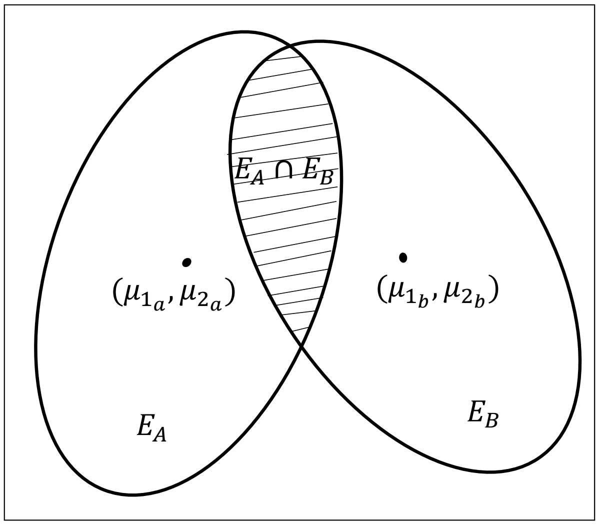


**Figure 32.—**Intersection of two 2-dimensional ellipsoids (ellipses)

Now, if these two ellipsoids $E_A$ and $E_B$ intersect with each other as depicted in Fig. 32, then every point within the intersection, $x \in E_A \cap E_B$ satisfies

$$\lambda(\mathbf{x} - \mathbf{a})^{\mathrm{T}}\mathbf{A}(\mathbf{x} - \mathbf{a}) + (1 - \lambda)(\mathbf{x} - \mathbf{b})^{\mathrm{T}}\mathbf{B}(\mathbf{x} - \mathbf{b}) \leq 1 \tag{48}$$

for $\lambda \in [0, 1]$. Note that Eq. (48) describes $\mathcal{E}_A$ and $\mathcal{E}_B$ for $\lambda = 0$ and $\lambda = 0$, respectively. To that end, we consider the following Lemmas proposed in Ref. 28.

**Lemma 7.1.** *Ref. 28 Given two ellipsoids $E_A(\mathbf{a}, \mathbf{A})$ and $E_B(\mathbf{b}, \mathbf{B})$ as defined in Eq.* (47)*, the set of points $x \in E_A \cap E_B$ satisfying Eq.* (48) *is either empty or one single point, or an ellipsoid, $\hat{E}_\lambda := E\left(\mathbf{m}, \hat{\mathbf{E}}_\lambda^{-1}\right)$ where*

$$\begin{aligned} \hat{\mathbf{E}}_\lambda &= \mathbf{E}_\lambda / k(\lambda), \quad \mathbf{E}_\lambda = \lambda\mathbf{A} + (1 - \lambda)\mathbf{B} \\ k(\lambda) &= 1 - \lambda\mathbf{a}^T\mathbf{A}\mathbf{a} - (1 - \lambda)\mathbf{b}^T\mathbf{B}\mathbf{b} + \mathbf{m}^T\mathbf{E}_\lambda\mathbf{m} \\ \mathbf{m} &= \mathbf{E}_\lambda^{-1}\left(\lambda\mathbf{A}\mathbf{a} + (1 - \lambda)\mathbf{B}\mathbf{b}\right). \end{aligned} \tag{49}$$

**Lemma 7.2.** *Ref. 28 Consider two ellipsoids $E_A(\mathbf{a}, \mathbf{A})$ and $E_B(\mathbf{b}, \mathbf{B})$ as defined in Eq.* (47)*. Then the ellipsoids $E_A$ and $E_B$ share no common point if and only if $\lambda^* \in (0, 1)$ exists with $k(\lambda^*) < 0$ where*

$$k(\lambda^*) := \min_\lambda k(\lambda), \quad \lambda \in (0, 1) \tag{50}$$

*and $k(\lambda)$ is as defined in Eq.* (49)*.*

### 7.2.2 Examples

To demonstrate this, we consider two Gaussian $3 - \sigma$ confidence ellipsoids in 2 dimensions (say $x$ and $y$) denoted by $E_{A_1}(^-{}_{A_1}, \mathbf{R}_{A_1})$ and $E_{B_1}(^-{}_{B_1}, \mathbf{R}_{B_1})$ where $^-{}_{A_1} = [2, 3]^{\mathrm{T}}$ and $^-{}_{B_1} = [7, 2]^{\mathrm{T}}$ are the mean (centre) and $\mathbf{R}_{A_1} = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 2 \end{bmatrix}$ and $\mathbf{R}_{B_1} = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 2 \end{bmatrix}$ are the correlation matrices of the $3-\sigma$ confidence ellipsoids (shape matrices). Using Eq. (50), we compute $k(\lambda^*) = 0.27415$. Figure 33 illustrates the $3-\sigma$ confidence ellipsoids $E_{A_1}$ and $E_{B_1}$. As remarked by Lemma 7.2, the two ellipsoids intersect with each other since $k(\lambda^*) > 0$.

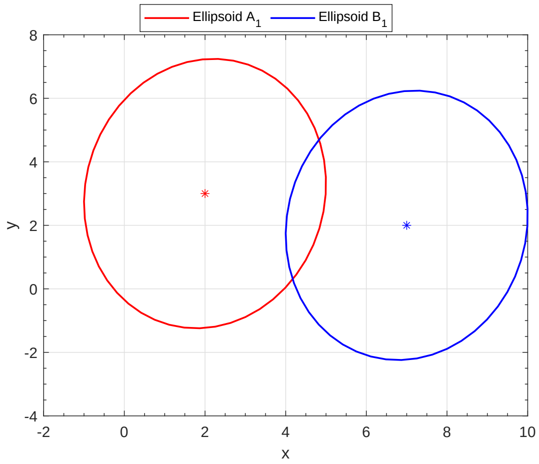


**Figure 33.**—$3-\sigma$ confidence ellipsoids $E_{A_1}$ and $E_{B_1}$ with $k(\lambda^*) = 0.27415$

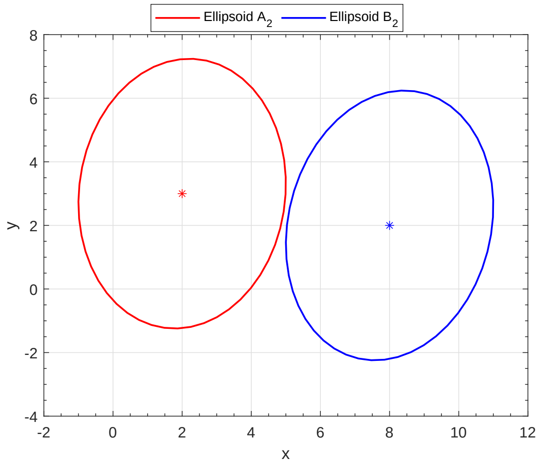


**Figure 34.**—$3-\sigma$ confidence ellipsoids $E_{A_2}$ and $E_{B_2}$ with $k(\lambda^*) = -0.0507$

Further, we consider two other Gaussian $3-\sigma$ confidence ellipsoids $E_{A_2}(\bar{}_{A_2}, \mathbf{P}_{A_2})$ and $E_{B_2}(\bar{}_{B_2}, \mathbf{P}_{B_2})$. For these ellipsoids, we assume $\bar{}_{A_2} = [2, 3]^{\mathrm{T}}$ and $\bar{}_{B_2} = [8, 2]^{\mathrm{T}}$ and $\mathbf{P}_{A_2} = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 2 \end{bmatrix}$ and $\mathbf{P}_{B_2} = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 2 \end{bmatrix}$. Using Eq. (50), we compute $k(\lambda^*) = -0.0507 < 0$ and from Fig. 34, it can be inferred that the two ellipses share no common point as described by Lemma 7.2.

### 7.2.3 Maximum Distribution of Uncertainties for No-conflict Condition

For the problem described in Section 7.1 with a traffic of sUAVs $i = 1, 2, \ldots, M$, we can define the scalar $k(\lambda, t)$ between the two $3 - \sigma$ confidence ellipses $E_{3\sigma_i}(t)$ and $E_{3\sigma_j}(t)$ for two sUAVs $i$ and $j$, $i, j = 1, 2, \ldots, M$ $(i \neq j)$ using Eq. (49) at any time instant $t$. Similarly, one can compute $k(\lambda^*, t)$ solving Eq. (50) at any instant $t$. We compute the maximum distribution of uncertain parameters for no-conflict conditions for the following two cases: (i) normally distributed uncertainties and (ii) uniformly distributed uncertainties.

### 7.2.4 Case I: Normally distributed uncertainties

For the case where the uncertain parameter vector $\mathbf{z} = [z_1, z_2, \ldots, z_d]^{\mathrm{T}}$ is normally distributed such that $z_i \in \mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})$, maximizing the distribution of uncertainties is equivalent to maximizing the standard deviation of the individual parameters $\sigma_{z_i}$. Let us denote $\boldsymbol{\sigma}_{\mathbf{z}} = [\sigma_{z_1}, \sigma z_2, \ldots, \sigma_{z_d}]^{\mathrm{T}}$. To that end, the maximum distribution of uncertainties for the no-conflict condition between any two sUAVs for all time instants $0 < t < \cdots < T$ is obtained upon solving the following optimization procedure

$$\begin{aligned} &\text{maximize } \boldsymbol{\sigma}_{\mathbf{z}} \\ &\text{subject to } k(\lambda^*, t) < 0, \quad \text{for all } t \in [0, T] \end{aligned} \tag{51}$$

The optimal solution to the no-conflict condition for normally distributed parameters is the probability density function of parameters, $z_i$ for all $i = 1, 2, \ldots, d$ given by $\mathcal{N}(\mu_{z_i}, \sigma^{2^*}_{z_i})$ where $\sigma^*_{z_i}$ is the solution to Eq. (51).

### 7.2.5 Case II: Uniformly distributed uncertainties

Given an uncertain parameter vector $\mathbf{z} = [z_1, z_2, \ldots, z_d]^{\mathrm{T}}$, let us now consider the case where each $z_i$ is uniformly distributed between a lower bound $\underline{z}_i$ and upper bound $\bar{z}_i$, denoted by $z_i \sim \mathcal{U}[\underline{z}_i, \bar{z}_i]$. To that end, we can compute the mean of the uniformly distributed parameter $z_i$ as follows

$$\mu_{z_i} = \frac{\underline{z}_i + \bar{z}_i}{2}$$

Since the filed plan of the sUAVs is predetermined, the mean of the uniformly distributed parameter $z_i$ is assumed to be known and equal to a deterministic value (based upon the filed plan). Now, we define $\delta$ as the deviation of lower and upper bounds from the mean as follows

$$\delta_i := \frac{\bar{z}_i - \underline{z}_i}{2}, \text{ such that } \underline{z}_i = \mu_{z_i} - \delta_i \text{ and } \bar{z}_i = \mu_{z_i} + \delta_i. \tag{52}$$

Therefore, for the case of uniformly distributed uncertainties, maximizing the distribution of uncertainties is equivalent to maximizing $\delta_i$. Let us denote $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_d]^{\mathrm{T}}$. To that end, the maximum distribution of uncertainties for the no-conflict condition between any two sUAVs for all time instants $0 < t < \cdots < T$ is obtained upon solving the following optimization procedure

$$\begin{aligned} &\text{maximize } \boldsymbol{\delta} \\ &\text{subject to } k(\lambda^*, t) < 0, \quad \text{for all } t \in [0, T] \end{aligned} \tag{53}$$

The optimal solution to optimization problem in Eq. (53) is denoted by $\boldsymbol{\delta}^*$ or equivalently $\delta^*_i$ for all $i = 1, 2, \ldots, d$. Further, the solution to the no-conflict condition for uniformly distributed parameters is the probability density function of parameters, $z_i$ for all $i = 1, 2, \ldots, d$ given by $\mathcal{U}(\underline{z}_i^*, \bar{z}_i^*)$ where $\underline{z}_i^*$ and $\bar{z}_i^*$ can be computed from Eq. (52) as follows

$$\underline{z}_i^* = \mu_{z_i} - \delta_i^* \text{ and } \bar{z}_i^* = \mu_{z_i} + \delta_i^*.$$

### 7.2.6 Time of Closest Approach

Consider two $3 - \sigma$ optimal confidence ellipses $E_{3\sigma_i}^*(t)$ and $E_{3\sigma_j}^*(t)$ of two sUAVs $i$ and $j$, $i, j = 1, 2, \ldots, M, i \neq j$ for all time instants $t \in [0, T]$ as a result of optimal distribution of parametric uncertainties (i.e. $z_i \sim \mathcal{N}(\mu_{z_i}, \sigma_{z_i}^{2^*})$ for the normally distributed case and $z_i \sim \mathcal{U}(\underline{z}_i^*, \bar{z}_i^*)$ for uniformly distributed case). Note that these optimal confidence ellipses satisfy $k(\lambda^*, t) < 0$ at all time instants $t \in [0, T]$. To that end, we define the time of closest approach between two sUAVs $i$ and $j$ as the time $t_c \in [0, T]$ at which the distance between two points on the optimal non-intersecting confidence ellipses $E_{3\sigma_i}^*(t)$ and $E_{3\sigma_j}^*(t)$ is the shortest. The time of closest approach, $t_c$ between any two vehicles can be computed by calculating the time at which $k(\lambda^*, t) < 0$ is maximum as follows

$$t_c = \arg\max_t k(\lambda^*, t). \tag{54}$$

### 7.2.7 Discussion of the optimization problems

Note that, the optimization problems in Eqs. (51) and (53) are vector-valued optimization problems and can also be cast as multi-objective optimization problems. In both the optimization problems in Eqs. (51) and (53), one needs to solve for the constraint of $k(\lambda^*, t) < 0$ which requires the mean vector $\boldsymbol{\mu}_{XY_i}(t)$ and covariance matrix $\mathbf{P}_i(t)$ at all time instants $t \in [0, T]$. For the nonlinear system models described in Eqs. (27) and (38), it is non-trivial to analytically compute the mean vector and covariance matrix of the North-East position at all time instants $t \in [0, T]$. Therefore, one needs to consider a computationally efficient uncertainty propagation technique. In this work, we use the framework of generalized polynomial chaos expansion to carry out uncertainty propagation, which is discussed earlier in Section 2.2.

## 7.3 Numerical Results

In this section, we apply the proposed framework in Section 7.2 and the gPC expansion-based uncertainty propagation technique to the problem of the traffic of sUAVs. To that end, we consider a traffic of $M = 2$ sUAVs with uncertainties in some of the parameters.

As stated earlier in Section 7.1, we consider the motion of the sUAVs at constant-altitude of $H_1(t) = H_2(t) = 250$ m and constant flight path angle of $\gamma_1(t) = \gamma_2(t) = 0$ rad. The commanded yaw acceleration is assumed to be constant for both the sUAVs such that $a_{yc_1} = a_{yc_2} = 0.01g$. The initial North-East coordinates of the sUAVs are such that $X_1(0) = 0$ m, $Y_1(0) = 0$ m and $X_2(0) = 1000$ m, $Y_1(0) = 500$ m for sUAV-1 and sUAV-2.

Both the sUAVs are assumed to have same initial specific energy $E_1(0) = E_2(0) = 270$ m, throttle position $\eta_1(0) = \eta_2(0) = 1$, pitch acceleration $a_{p_1}(0) = a_{p_2}(0) = g$, and yaw acceleration $a_{y_1}(0) = a_{y_2}(0) = 0.01g$. The initial heading of the sUAV-1 is assumed to be $\chi_1(0) = 0$ radians and that of sUAV-2 is assumed to be $\chi_2(0) = 0.15$ radians. Further, for both the sUAVs, $\lambda_n = \lambda_p = \lambda_y = 0.15$, $g = 9.81$ m/s$^2$, and $\rho = 1.225$ Kg/m$^3$. The total flight time for both the sUAVs is assumed to be $T = 10$ minutes.

We assume uncertainties in the zero-lift drag coefficient, wing planform area, and maximum available thrust. The parametric uncertainties are assumed to be the same for both sUAVs and are assumed to follow uniform distribution as follows:

$$\begin{aligned} C_{D0} &\sim \mathcal{U}[C_1, C_2] \\ S &\sim \mathcal{U}[S_1, S_2] \\ T_{\text{max}} &\sim \mathcal{U}[T_1, T_2] \end{aligned} \tag{55}$$

For the uniformly distributed parametric uncertainties, we define the mean value and dispersion as follows

$$\begin{aligned} \mu_C &= \frac{C_1 + C_2}{2}, \quad \delta_C = \frac{C_2 - C_1}{2} \\ \mu_S &= \frac{S_1 + S_2}{2}, \quad \delta_S = \frac{S_2 - S_1}{2} \\ \mu_T &= \frac{T_1 + T_2}{2}, \quad \delta_T = \frac{T_2 - T_1}{2} \end{aligned} \tag{56}$$

Now, the problem of finding the maximum distribution of uncertainties is recast to the problem of computing maximum values of $\delta_C$, $\delta_S$, and $\delta_T$ for the no-conflict condition between two sUAVs with known mean values of uncertainties as given below

$$\begin{aligned} \mu_C &= 0.045 \\ \mu_S &= 0.65 \text{ m}^2 \\ \mu_T &= 6 \text{ N} \end{aligned} \tag{57}$$

We employ $4^{th}$ order (i.e. $P = 4$) of the gPC expansion technique to carry out uncertainty propagation. Further, to compute the coefficients of gPC expansion, the pseudospectral collocation technique is used. To obtain the collocation nodes and weights for the pseudospectral collocation, a multi-dimensional sparse grid-based quadrature technique (Ref. 40) with 31 quadrature points ($N_q = 31$) is utilized. We solve the multi-objective optimization problem in as a goal-attainment problem using the 'fgoalattain' routine in MATLAB.

### 7.3.1 Simulation Results

Upon solving the optimization routine, we obtain the following maximum values of $\delta_C$, $\delta_S$, and $\delta_T$ for the no-conflict condition between two sUAVs

$$\begin{aligned} \delta_C^* &= 3.0374 \times 10^{-5} \\ \delta_S^* &= 0.5123 \text{ m}^2 \\ \delta_T^* &= 0.4969 \text{ N}. \end{aligned} \tag{58}$$

Now, using Eq. (52), the maximum probability distributions of the uncertain parameters for no-conflict condition are obtained to be as follows

$$\begin{aligned} C_{D0}^* &\sim \mathcal{U}[0.04496, 0.04503] \\ S^* &\sim \mathcal{U}[0.6371, 0.6629] \text{ m}^2 \\ T_{\text{max}}^* &\sim \mathcal{U}[5.5039, 6.4961] \text{ N}. \end{aligned} \tag{59}$$

With the calculated maximum distribution of uncertainties in Eq. 59, we carry out gPC expansion-based uncertainty propagation and compute the mean and $3 - \sigma$ confidence ellipses of two sUAVs. Figure 35 illustrates the temporal variation of the mean and $3 - \sigma$ confidence ellipses of the two sUAVs (denoted as sUAV-1 and sUAV-2) along north and east. Further, Fig. 36 depicts the mean and $3-\sigma$ confidence ellipses of the sUAVs' position along north and east at various time instants. Note that the $3-\sigma$ confidence ellipses of two sUAVs are non-intersecting at all time instants throughout the flight time. Moreover, we calculate the time of the closest approach to be $t_c = 5.4$ minutes using Eq. (54). The position mean and $3-\sigma$ confidence ellipses of the two sUAVs at the time of closest approach are illustrated in Fig. 37. Clearly, at the time of closest approach, the two confidence ellipses are non-intersecting. Therefore, the calculated maximum distribution of uncertainties in Eq. 59 results in a no-conflict condition in the two sUAVs during the flight time.

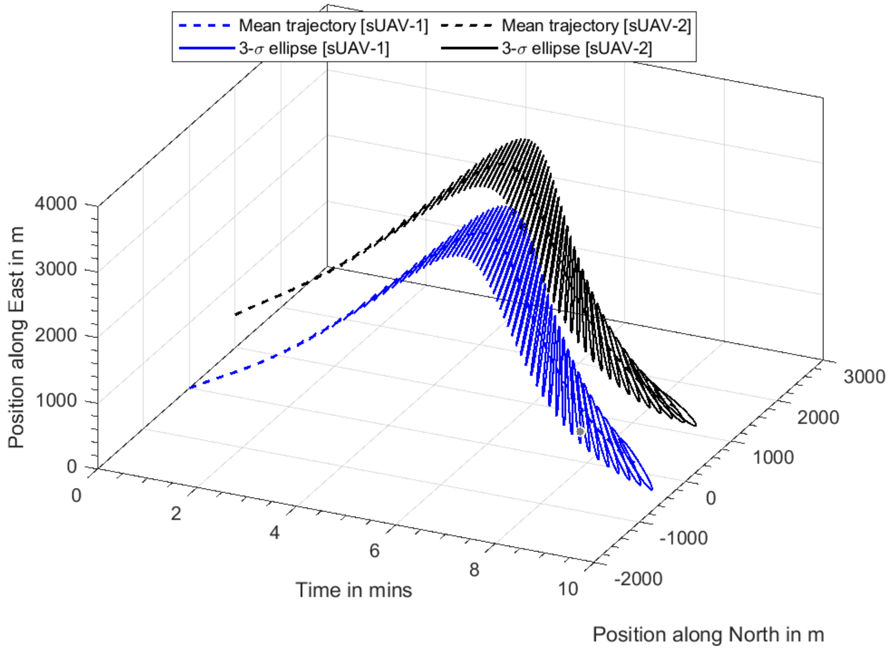


**Figure 35.—**Mean trajectory and 3-$\sigma$ confidence ellipses of sUAVs' position

### 7.3.2 Risk Analysis with larger distribution of uncertainties

To demonstrate the accuracy of our approach, we carry out gPC expansion-based uncertainty propagation with distribution of $T_{\text{max}}$ larger than that in Eq. 59 as follows

$$T_{\text{max}} = \mathcal{U}[5.5, 6.5]\ \text{N}, \quad \delta_T = 0.5\ \text{N}, \quad \mu_T = 6\ \text{N} \tag{60}$$

Note that the mean $\mu_T$ of the uniform distribution is kept equal to the mean used to calculate the maximum distribution of $T_{\text{max}}$ for no-conflict condition. On the other hand, the deviation of the upper and lower bounds from the mean, i.e. $\delta_T$ is increased by a small amount of $0.0031$ N. The distribution of $C_{D0}$ and $S$ are assumed to be the same as that in Eq. 59, i.e. $C^*_{D0} \sim \mathcal{U}[0.04496, 0.04503]$ and $S^* \sim \mathcal{U}[0.6371, 0.6629]$ m$^2$.

The position mean and $3-\sigma$ confidence ellipses of the two sUAVs at the $t = 5.4$ minutes is illustrated in Fig. 38. Clearly, the two confidence ellipses of the sUAVs are intersecting. Note that, even a small increment

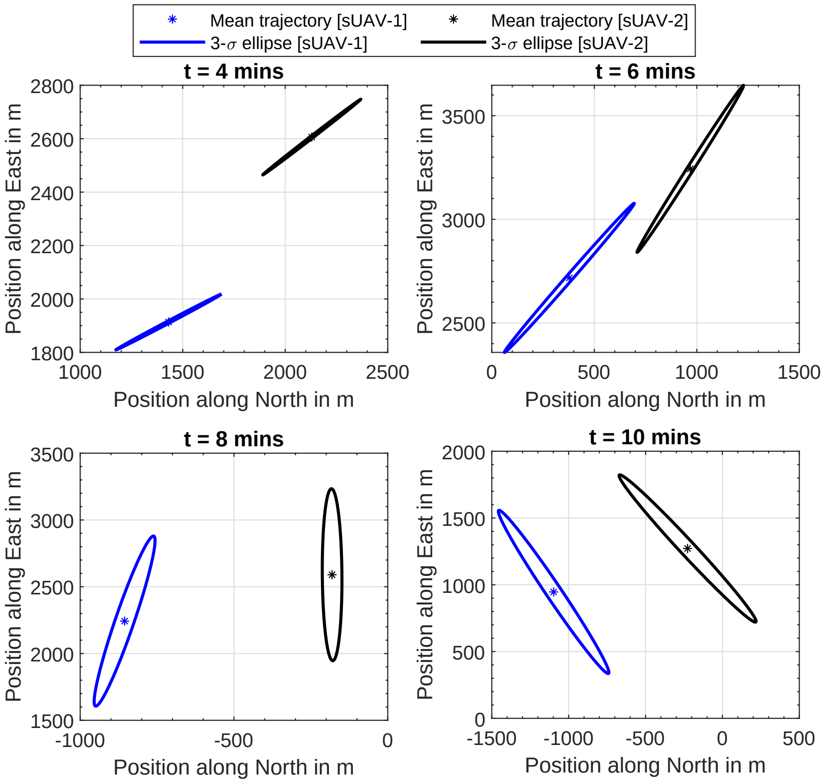


**Figure 36.—**Position mean and $3 - \sigma$ confidence ellipses of two sUAVs at various time instants

of 0.0031 N in $\delta_T$ from the obtained optimal $\delta_T^*$ in Eq. 58 results in possibility of conflict between the two sUAVs. This verifies the accuracy of the proposed approach.

# 8 Ground Impact Risk (Crash Region) due to Loss of Control Resulting from Actuator Failure

In this section, we use the gPC expansion technique to analyze the risk of casualty to non-participants on the ground as a result of loss of control in a group of unmanned aerial vehicles. We model the loss of control as a result of actuator failure in fixed-wing UAVs and rotary-wing vehicles.

## Risk due to actuator failure in fixed-wing sUAV

The loss of control in the fixed-wing sUAV is realized as follows:

(i) failure in throttle position of the sUAV such that

$$\eta = 0, \qquad \text{for } t > 5 \text{ mins} \qquad \text{(see Eq. 28)} \tag{61}$$

(ii) failure in the pitch stick (elevator deflection) of the sUAV such that

$$a_p = 0, \qquad \text{for } t > 5 \text{ mins} \qquad \text{(see Eq. 27)} \tag{62}$$

The numerical simulations consider a typical geographical region in the City of Arlington, TX. Further, we consider uncertainties in the initial position of the sUAV such that, $X(0) \sim \mathcal{N}(0, 1^2)$ m and $Y(0) \sim$

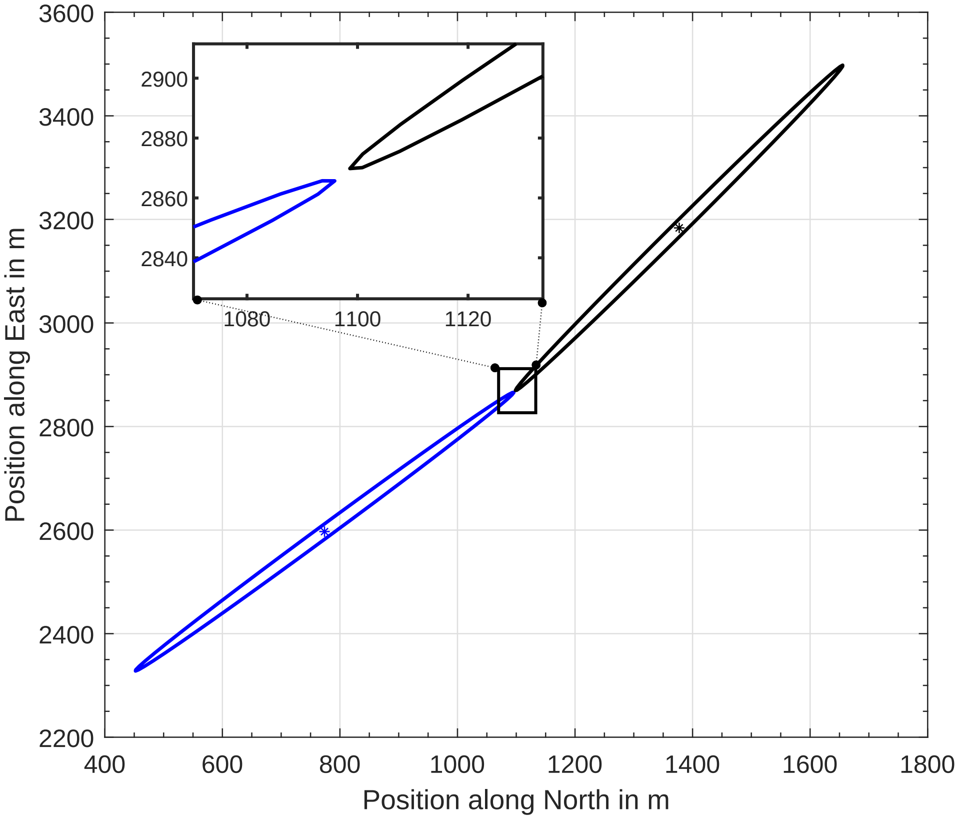


**Figure 37.**—Position mean and $3-\sigma$ confidence ellipses of two sUAVs at the time of closest approach $t_c = 5.4$ mins

$\mathcal{N}(0, 1^2)$ m. The initial altitude is considered to be 20 m and the aircraft is assumed to have an altitude and heading hold PI autopilot. The parametric uncertainties in the sUAV are listed in Table 17.

**Table 17.**—Parametric uncertainties

| Parameters | PDF |
|---|---|
| $m$ | $\mathcal{U}[2, 4.25]$ Kg |
| $C_{D0}$ | $\mathcal{U}[0.0035, 0.006]$ |
| $S$ | $\mathcal{U}[0.4, 0.625]$ $m^2$ |
| $k$ | $\mathcal{U}[0.0075, 0.0325]$ |
| $T_{\max}$ | $\mathcal{U}[3.75, 6.25]$ N |

The possible trajectories of the sUAV are simulated by considering the mixed sparse grid gPC expansion technique. Due to the failure in the throttle position and pitch stick after 5 mins, the sUAV is bound to crash. Figures 39 and 40 depict the possible crash site density of the sUAV. The North-East coordinates of the sUAV are converted to the geo-latitude and geo-longitude to demonstrate the impact region. From Figs. 39 and 40, we can infer that the size of uncertainty in the crash location is small, implying high robustness of the controller to uncertainties in the sUAV's initial position and its parameters. This signifies that the impact

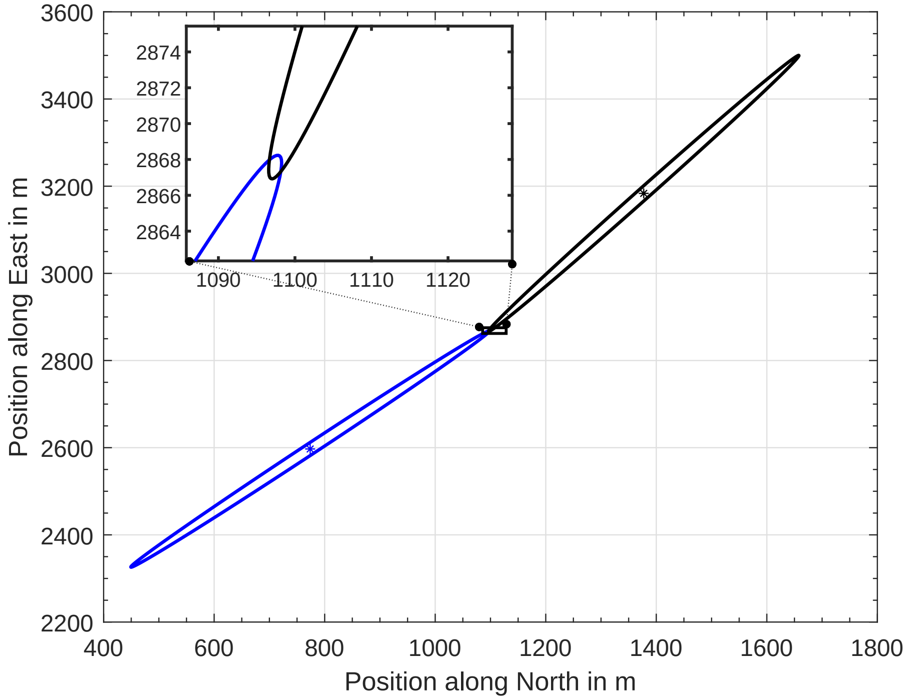


**Figure 38.—**Position mean and $3 - \sigma$ confidence ellipses of two sUAVs at $t = 5.4$ mins (with larger distribution of uncertainties)

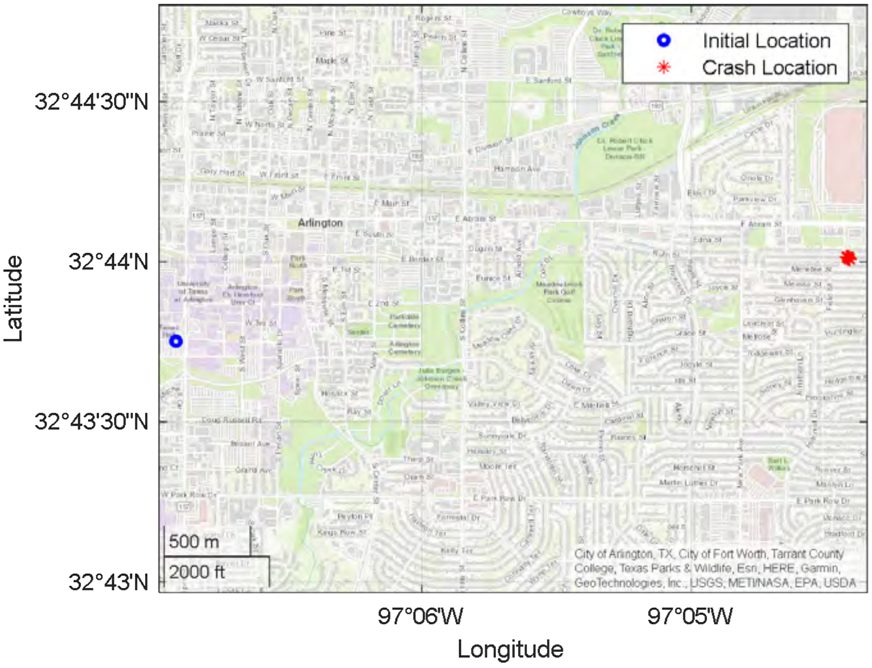

Initial Location
Crash Location
Arlington
Latitude
32°44'30"N
32°44'N
32°43'30"N
32°43'N
97°06'W
97°05'W
Longitude
500 m
2000 ft
City of Arlington, TX, City of Fort Worth, Tarrant County College, Texas Parks & Wildlife, Esri, HERE, Garmin, GeoTechnologies, Inc., USGS, METI/NASA, EPA, USDA

**Figure 39.—**Initial and crash-site geo-density plot with sUAV uncertainties

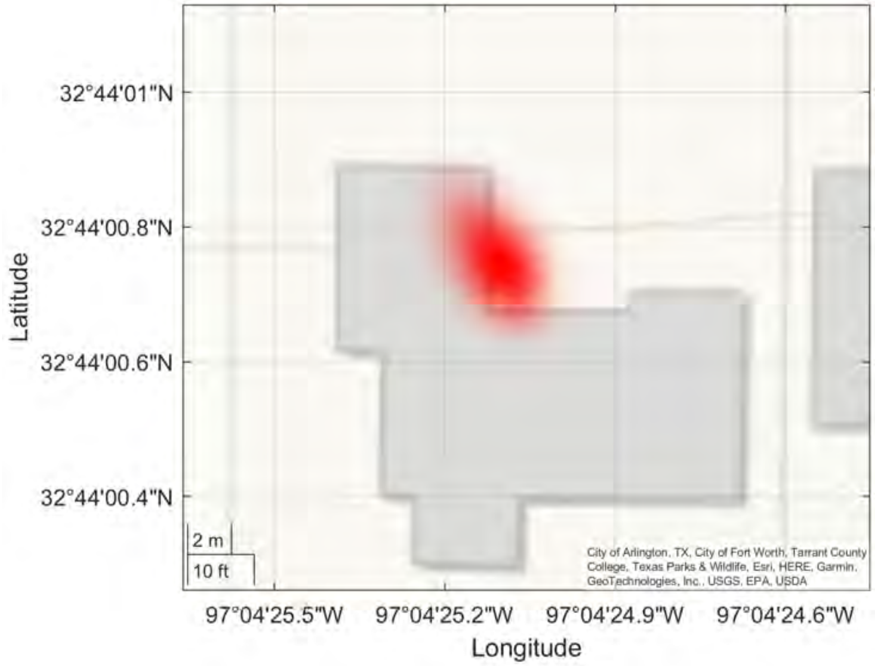

32°44'01"N
32°44'00.8"N
32°44'00.6"N
32°44'00.4"N
Latitude
97°04'25.5"W
97°04'25.2"W
97°04'24.9"W
97°04'24.6"W
Longitude
2 m
10 ft
City of Arlington, TX, City of Fort Worth, Tarrant County College, Texas Parks & Wildlife, Esri, HERE, Garmin, GeoTechnologies, Inc., USGS, EPA, USDA

**Figure 40.—**Geo-density plot of the crash site with sUAV uncertainties

footprint is small due to actuator failure in the presence of uncertainties. Further, the bivariate PDF in Fig. 40 looks elongated, i.e. the resemblance to a bivariate Gaussian is small, reason being the occurrence of a mixture of Uniform uncertainties (parameters - Table 17) and Gaussian uncertainties (initial position).

Further, we also consider the scenario where the fixed-wing UAV operates in an uncertain wind field, in addition to uncertain initial position and parameters. This uncertain wind field is characterized in Table 11. The geo-density plot of the crash site in the presence of uncertainties and actuator failure is depicted in Figs. 41 and 42. It can be observed that the expanse of the crash site is larger than the case with only sUAV uncertainties. Thus, with actuator failures, the impact region of a ground collision is larger with uncertainties in both the sUAV and the external wind field.

## Risk due to actuator failure in multiple quadrotors

Next, we consider the traffic of quadrotors in an uncertain wind field. The model for the quadrotors is described in Eq. 38. Further, we utilize attitude, heading, and speed-hold PD autopilots. The initial geo-coordinates of the four quadrotors are depicted in Fig. 43.

To demonstrate the actuator failure, we simulate a rotor failure for each quadrotor as follows:

(i) For quadrotor-1, $\omega_2 = 0$ for $t > 8$ minutes,

(ii) For quadrotor-2, $\omega_2 = 0$ for $t > 7$ minutes,

(iii) For quadrotor-3, $\omega_2 = 0$ for $t > 6$ minutes,

(iv) For quadrotor-4, $\omega_2 = 0$ for $t > 10$ minutes.

Since, a rotor fails in all of the four quadrotors at various time instants, the altitude hold, velocity tracking, and heading tracking autopilots fail to perform their desired tasks. This results in loss of control and further, the quadrotors begin to lose altitude, resulting in crashes at four different locations. Due to

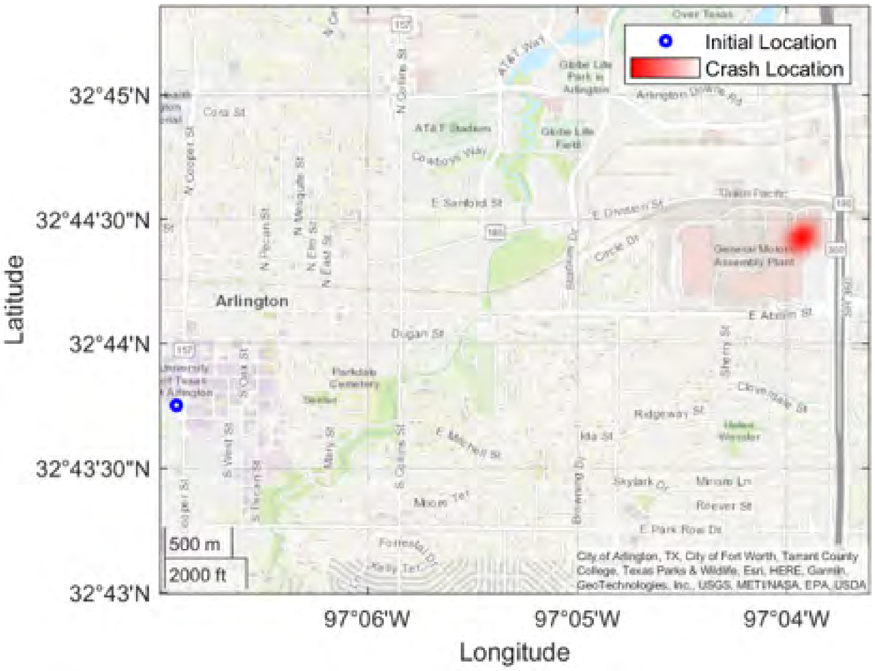

Initial Location
Crash Location
Latitude
32°45'N
32°44'30"N
32°44'N
32°43'30"N
32°43'N
Longitude
97°06'W
97°05'W
97°04'W
Arlington
AT&T Stadium
Globe Life Park in Arlington
Globe Life Field
General Motors Assembly Plant
500 m
2000 ft
City of Arlington, TX, City of Fort Worth, Tarrant County College, Texas Parks & Wildlife, Esri, HERE, Garmin, GeoTechnologies, Inc., USGS, METI/NASA, EPA, USDA

**Figure 41.—**Initial and crash-site geo-density plot with sUAV and wind uncertainties

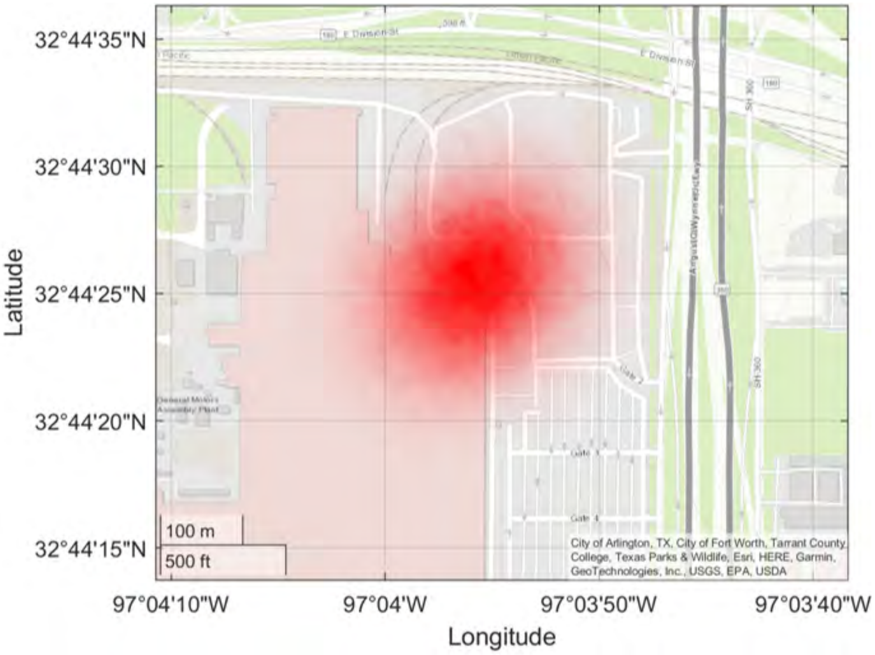

Latitude
32°44'35"N
32°44'30"N
32°44'25"N
32°44'20"N
32°44'15"N
Longitude
97°04'10"W
97°04'W
97°03'50"W
97°03'40"W
100 m
500 ft
City of Arlington, TX, City of Fort Worth, Tarrant County College, Texas Parks & Wildlife, Esri, HERE, Garmin, GeoTechnologies, Inc., USGS, EPA, USDA

**Figure 42.—**Geo-density plot of the crash site with sUAV and wind uncertainties

wind uncertainties, a particular crash location can not be exactly computed. However, we can obtain the PDF of the crash locations (possible locations) using the gPC expansion technique. Based on the PDF of the location of the crash, we estimate the impact risk. Figures 44, 45, 46, and 47 depict the geo-density maps with the estimated locations of the quadrotors at various time instants.

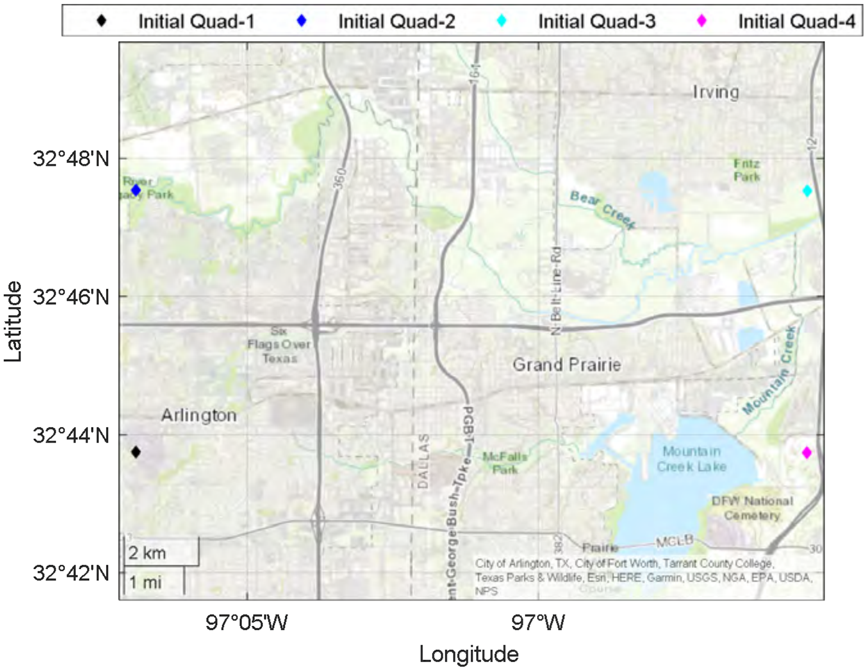


**Figure 43.—**Geographical map with initial conditions of quadrotors in the cities of Arlington, Fort Worth, Grand Prairie, and Irving in Texas

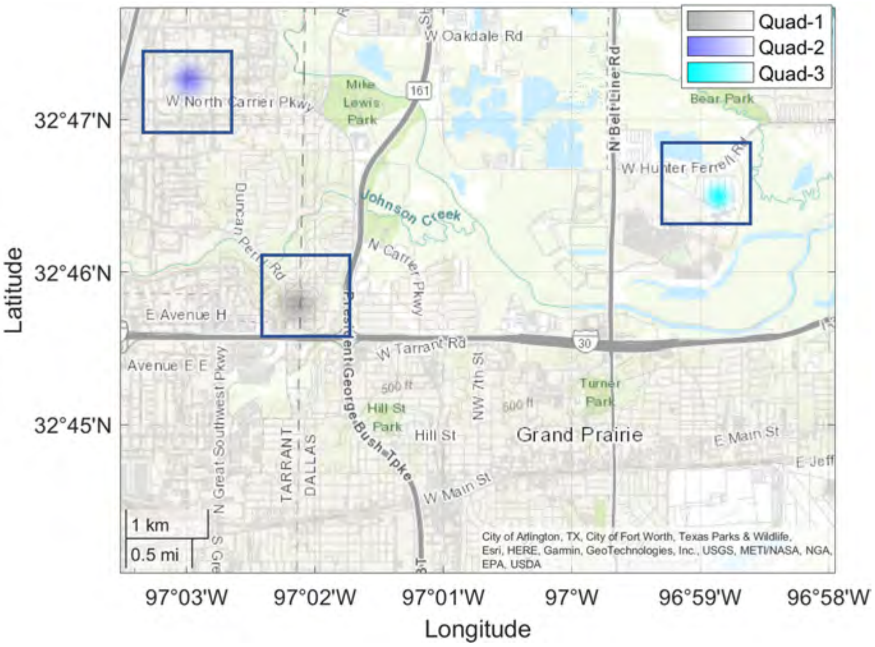


**Figure 44.—**Geographical map with estimated crash region's PDF at t = 6.03 minutes

It was observed that quadrotor-3 crashed first at around 6.03 minutes, followed by quadrotor-2 at around 7.03 minutes, quadrotor-1 at around 8.03 minutes, and finally quadrotor-4 at around 10.03 minutes. The individual possible crash location (zoomed plot) for the quadrotor is illustrated in Figs. 48, 49, 50, and 51.

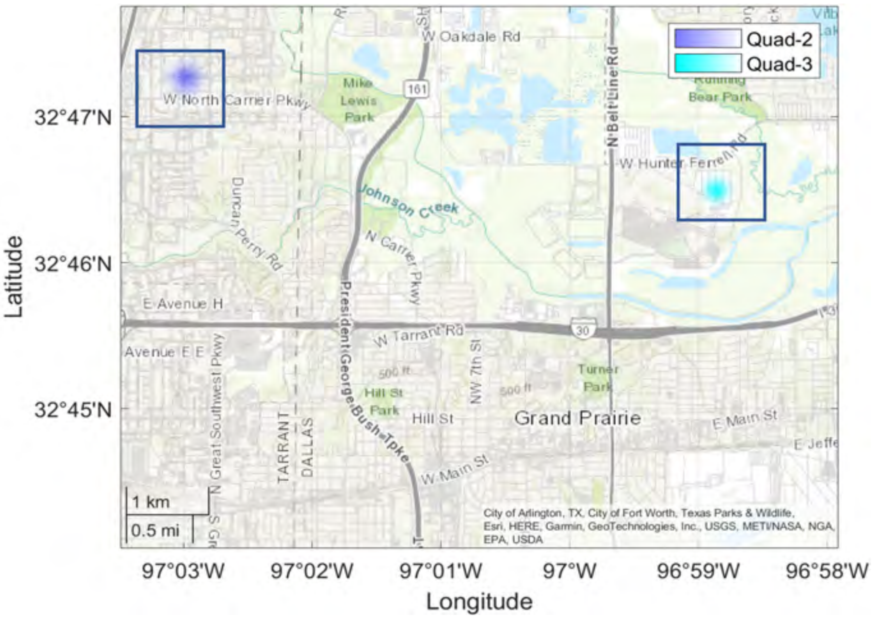


**Figure 45.**—Geographical map with estimated crash region's PDF at t = 7.03 minutes

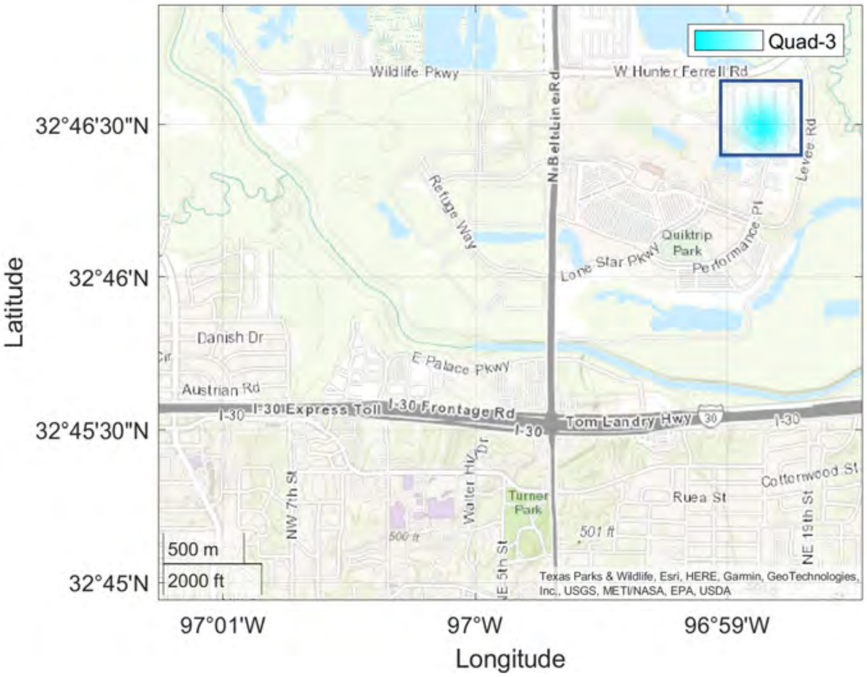


**Figure 46.**—Geographical map with estimated crash region's PDF at t = 8.03 minutes

This specific study of the impact risk provides another layer that can be combined with other tools under development for safety of emerging aviation operations, and be used to estimate the risk of casualty to non-participants on the ground. Such prediction would entail utilization of population density data and/or property information that could be under risk from the crash/impact.

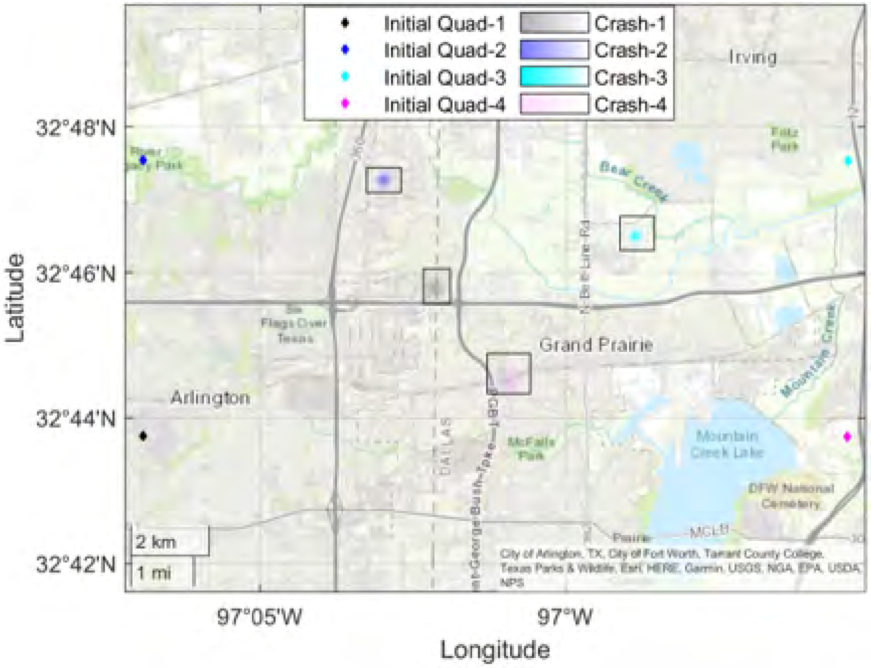


**Figure 47.—**Geographical map with estimated crash region's PDF at t = 10.03 minutes

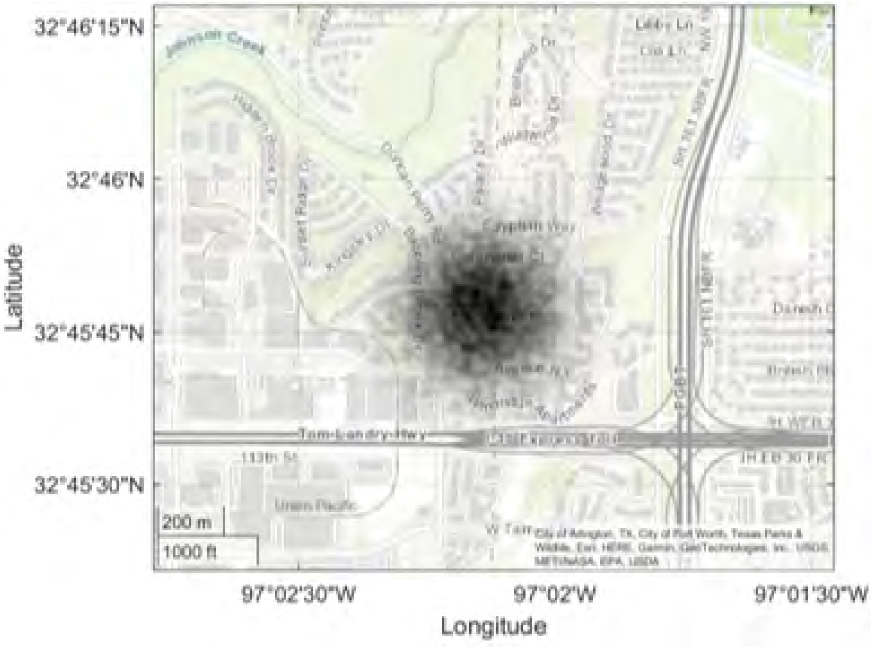

32°46'15"N
32°46'N
32°45'45"N
32°45'30"N
Latitude
97°02'30"W
97°02'W
97°01'30"W
Longitude
Johnson Creek
Tom Landry Hwy
Union Pacific
200 m
1000 ft
City of Arlington, TX, City of Fort Worth, Texas Parks & Wildlife, Esri, HERE, Garmin, GeoTechnologies, Inc., USGS, METI/NASA, EPA, USDA

**Figure 48.—**Geographical map with the estimated crash region for quadrotor-1

# 9 Learning Stochastic Processes Using Gaussian Processes

Gaussian processes have received significant attention in recent years, because of their simplicity, flexibility, and feasibility in solving stochastic problems which are currently dominated by other complicated

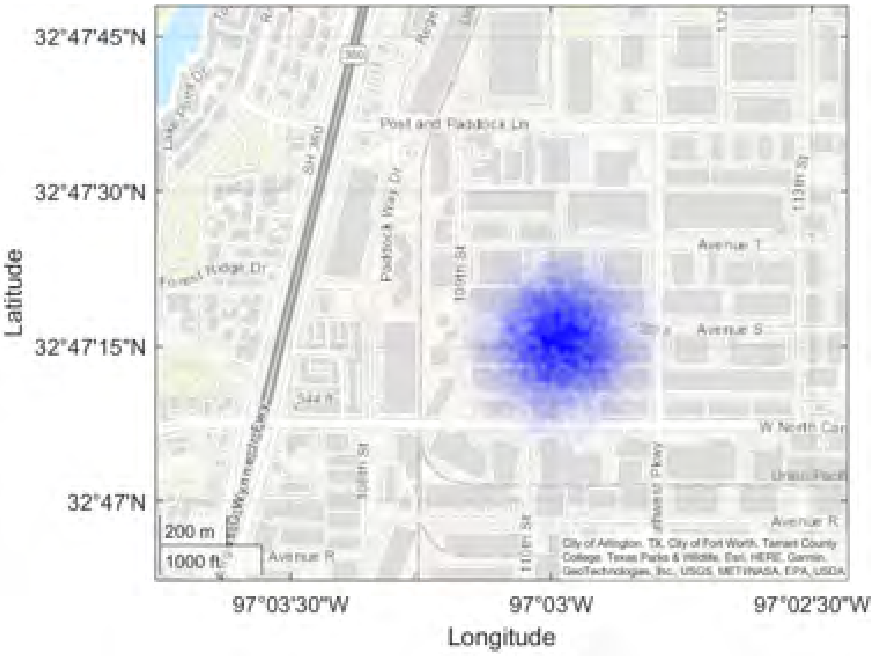


**Figure 49.—**Geographical map with the estimated crash region for quadrotor-2

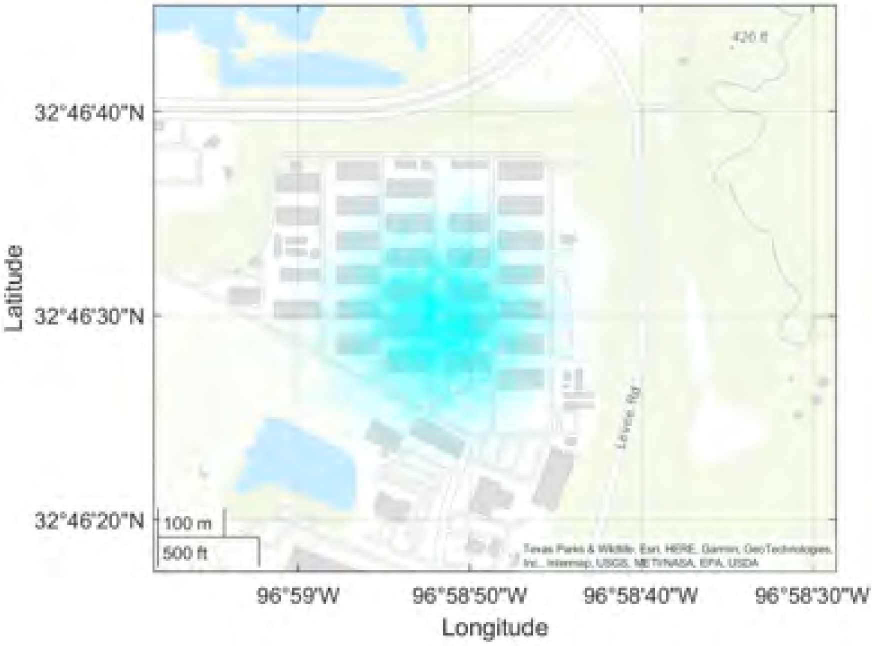


**Figure 50.—**Geographical map with the estimated crash region for quadrotor-3

techniques such as neural networks. In the literature, most of the predictions for dynamic systems are conducted by studying their mathematical models. However, developing a mathematical model that accounts for uncertainties in the system is not always feasible. In addition, solving mathematical models involving stochastic differential equations is computationally cumbersome, especially when the underlying system is

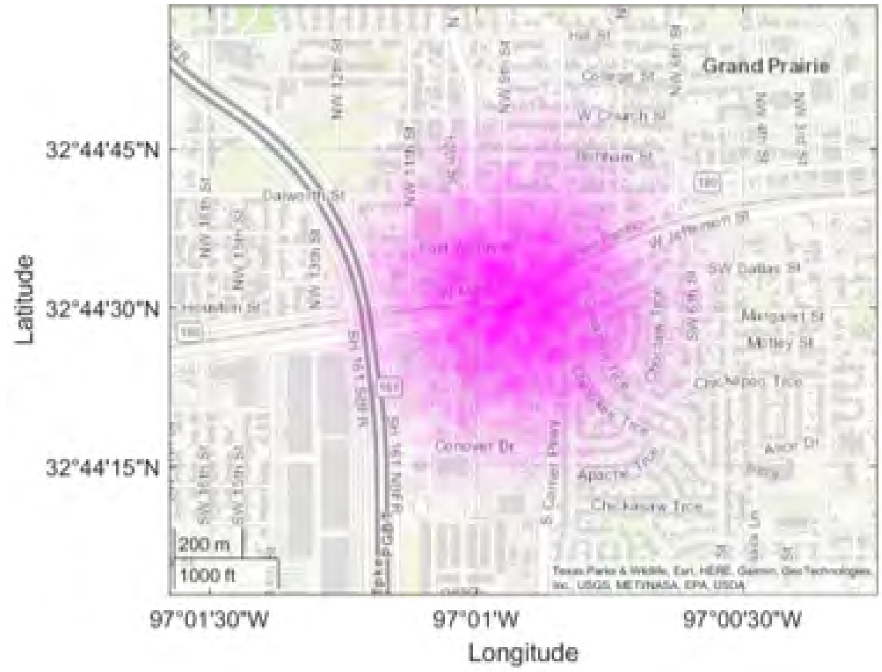


**Figure 51.—**Geographical map with the estimated crash region for quadrotor-4

governed by a large number of uncertain variables. As an alternative to solving stochastic differential equations, learning stochastic processes has been marginally researched in the literature using machine learning techniques. In general, machine learning refers to methods or algorithms that can ”learn” from available information or data enabling them to carry out desired tasks Ref. 41. In the context of machine learning, supervised learning involves predicting outputs from a system corresponding to a given set of inputs based on the knowledge of prior sets of inputs and outputs. In the literature, a significant amount of research has been conducted on learning stochastic processes. Tanevski et al. Ref. 42 studied learning stochastic dynamical systems using prior information for application to problems in biology. Jiang et al. Ref. 43 utilized deep learning to determine the damping coefficient of a random system.

There are different supervised learning techniques used in engineering applications, out of which, the Gaussian processes model is a probabilistic machine learning framework that has been widely used for regression and classification tasks. Gaussian processes have earlier been used for predictive control (Ref. 44), uncertainty quantification in aerodynamics (Ref. 45), time series prediction (Ref. 46), and travel time prediction (Ref. 47). Moreover, Amer et al. (Ref. 48) applied Gaussian process regression integrated into other methods in structural health monitoring for detecting probabilistic damage in a structure, Mohanty et al. Ref. 49 implemented Gaussian process for predicting the fatigue life of material and Lee et al. (Ref. 50) implemented machine learning techniques to predict aircraft taxi time.

In this section, we apply Gaussian processes for prediction and uncertainty propagation. For the prediction, we exemplify the technique to predict delays in commercial air carrier flights in the national airspace. The commercial air vehicle domain is chosen because there is true flight data for which we can verify and validate the technique. We then implement the GPR technique to propagate the uncertainties in sUAV flight operations, the primary domain for this research effort.

## 9.1 Mathematical Preliminaries

This section reviews the basic mathematical concepts needed to understand Gaussian processes.

### 9.1.1 Gaussian Random Variable, Multivariate Distribution, and Gaussian Random Vectors

Multivariate normal distribution, also known as Gaussian distribution, is the foundation for Gaussian processes. Any random variable $Y$ is said to be Gaussian or normally distributed if its probability density function, $f_Y(y)$ is given by Ref. 51

$$f_Y(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2}\right), \text{ for all } y \in \mathbb{R}$$

where $\mu = \mathbb{E}[Y]$ is the mean or expected value and $\sigma^2$ is the variance of $Y$.

Multivariate distribution is a distribution of multiple correlated random variables. The correlation among the random variables is generally represented by a symmetric, positive semi-definite matrix, known as the covariance matrix. Let us denote the covariance matrix among $Y_i$, $i = 1, 2 \ldots, n$ random variables with $\mathbf{K} = \mathbf{K}^{\mathrm{T}} \in \mathbb{R}^{n \times n}$. For all $i, j = 1, 2, \ldots, n$, the $(i, j)$ element of $\mathbf{K}$ represents the covariance between $Y_i$ and $Y_j$ denoted by $\text{cov}[Y_i, Y_j] = \mathbb{E}\left[(Y_i - \mathbb{E}[Y_i])(Y_j - \mathbb{E}[Y_j])\right]$ such that

$$\mathbf{K} = \begin{bmatrix} \text{cov}[Y_1, Y_1] & \text{cov}[Y_1, Y_2] & \cdots & \text{cov}[Y_1, Y_n] \\ \text{cov}[Y_2, Y_1] & \text{cov}[Y_2, Y_2] & \cdots & \text{cov}[Y_2, Y_n] \\ \vdots & \vdots & \cdots & \vdots \\ \text{cov}[Y_n, Y_1] & \text{cov}[Y_n, Y_2] & \cdots & \text{cov}[Y_n, Y_n]. \end{bmatrix} \tag{63}$$

Now, we can define a Gaussian random vector $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_n]^{\mathrm{T}} \in \mathbb{R}^n$ if the distribution of $\mathbf{Y}$ is multivariate Gaussian whose probability density function is given by

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n \, |\mathbf{K}|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right) \tag{64}$$

where $\boldsymbol{\mu} = [\mathbb{E}[Y_1], \mathbb{E}[Y_2], \ldots, \mathbb{E}[Y_n]]^{\mathrm{T}}$ is the mean vector and $\mathbf{K}$ as defined in Eq. (63) is the covariance matrix of $\mathbf{Y}$. In Eq. (64), $|\cdot|$ denotes the determinant operator.

### 9.1.2 Gaussian Processes

Gaussian processes (GP) are stochastic processes that can be used to model continuous functions and in addition, are generalizations of the multivariate normal distribution to infinite dimensions (Ref. 52). A function $f : \mathbb{R}^d \to \mathbb{R}$ is a sample drawn from a GP given by Eq. (65), if the probability density function of $f$ at any input $\mathbf{y} \in R^d$ is jointly Gaussian.

$$f(\mathbf{y}) \sim GP(\mu(\mathbf{y}), \sigma(\mathbf{y}, \mathbf{y}')) \tag{65}$$

In Eq. (65), $\mu(\mathbf{y})$ = $\mathbb{E}[f(\mathbf{y})]$ is the mean function or the expected value of functions present in Gaussian process evaluated at $\mathbf{y}$ and $\sigma(\mathbf{y}, \mathbf{y}')$ is the covariance function given by

$$\sigma(\mathbf{y}, \mathbf{y}') = \mathbb{E}\left[(f(\mathbf{y}) - \mu(\mathbf{y}))\,(f(\mathbf{y}') - \mu(\mathbf{y}'))\right]$$

Note that, the covariance function $\sigma : \mathbb{R}^d \times \mathbb{R}^d$ maps two vector inputs to a scalar output. In the Gaussian process, kernel functions $k(\mathbf{y}, \mathbf{y}')$ is used to specify the covariance function (Ref. 53) and therefore capture the similarity between inputs $\mathbf{y}$ and $\mathbf{y}'$. To that end, a larger value for $k(\mathbf{y}, \mathbf{y}')$ corresponds to the high correlation between the GP outputs $f(\mathbf{y})$ and $f(\mathbf{y}')$ and thus, these outputs tend to be similar for both inputs

$\mathbf{y}$ and $\mathbf{y}'$ (Ref. 52). Further, the covariance matrix $\mathbf{K}$ associated with the covariance function is positive semi-definite (Ref. 54). We can select from a number of covariance functions and also combine existing covariance functions as required. The selection of covariance function depends on characteristics of the model such as how smooth is the model, how scattered are the model data, and how unpredictable is the model (Ref. 55).

One of the most common covariance functions is exponentiated quadratic, also known as squared exponential kernel which is given by

$$k(\mathbf{y}, \mathbf{y}') = \sigma^2 \exp\left(-\frac{1}{2}\left(\mathbf{y} - \mathbf{y}'\right)^{\mathrm{T}} \boldsymbol{\Lambda}^{-1}\left(\mathbf{y} - \mathbf{y}'\right)\right), \text{ for } \mathbf{y}, \mathbf{y}' \in \mathbb{R}^d$$

where $\sigma^2$ is the signal variance of the function $f$ and $\boldsymbol{\Lambda} = \text{diag}([l_1^2, l_2^2, \ldots, l_d^2])$ is a diagonal matrix of squared characteristic length-scales $l_i$, $i = 1, 2, \ldots, d$. For more details on different types of kernels, please refer to Section 9.3.

The accuracy of the Gaussian process depends on the selection of the covariance function and its hyperparameter because incorrect selection can give poor results. So, hyperparameters of kernels can be optimized using the maximum log marginal likelihood (Ref. 54). The log marginal likelihood is written as

$$\log p(\mathbf{x}|\mathbf{y}, \phi) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{d}{2}\log 2\pi$$

where $\mathbf{y} \in R^d$ is given data , $\phi$ are hyperparameters and $K$ is the covariance matrix of output $\mathbf{x}$. The terms in log marginal likelihood relation penalize different features (Ref. 56). The term $-\frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y}$ penalizes how well the hyperparameter values currently used, fit the data. Similarly, the term $-\frac{1}{2}\log|\mathbf{K}|$ is penalty for complexity and - $\frac{d}{2}\log 2\pi$ is a constant for normalization. The log marginal likelihood can be maximized by taking the partial derivative of the log marginal likelihood relation concerning $\phi$.

## 9.2 Gaussian Process Regression

In linear regression techniques, the main objective is to relate a function $f(y)$ to a specific model (e.g. $f(y) = my + c$)). In Gaussian process regression (GPR), the function $f(y)$ is not specific and allows data to define the model (Ref. 57). GPR is a non-parametric Bayesian approach where target variables are assumed to be random variables and extracted from a probability distribution. Also due to the non-parametric nature, the number of parameters typically increase with training data. That said, a non-parametric model though computationally complex, is very flexible for data fitting (Ref. 58). Let $\mathbf{y} = [y_1,\ y_2,\ \ldots,\ y_n]^T$ denote a vector of predictor variables and $\mathbf{x} = [x_1,\ x_2,\ \ldots,\ x_n]^T$ denote a vector of target variables. Also, $\mathbf{Y}$ denotes all related data known at present. Then, the posterior probability distribution of $\mathbf{x}$ after knowing $\mathbf{y}$ is specified by Bayes theorem as follows (Ref. 59):

$$P(\mathbf{x}|\mathbf{y}, \mathbf{Y}) = \frac{P(\mathbf{y}|\mathbf{x}, \mathbf{Y})P(\mathbf{x}|\mathbf{Y})}{P(\mathbf{y}|\mathbf{Y})}$$

$P(\mathbf{y}|\mathbf{x}, \mathbf{Y})$ represents the likelihood of data, $P(\mathbf{x}|\mathbf{Y})$ represents the prior probability and $P(\mathbf{y}|\mathbf{Y})$ is the normalization constant. This normalization constant is the marginal probability of $\mathbf{y}$ and doesn't depend on the target. The marginal probability is obtained from the joint probability by integrating the target. Let us consider a multivariate Gaussian distribution given by

$$f(\mathbf{Y}, \mathbf{X}) \sim N\left(\begin{bmatrix}\mu_Y \\ \mu_X\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{YY} & \mathbf{K}_{YX}^T \\ \mathbf{K}_{YX} & \mathbf{K}_{XX}\end{bmatrix}\right)$$

where $\mathbf{K}_{YY}$ is the variance of $\mathbf{Y}$, $\mathbf{K}_{XX}$ is the variance of $\mathbf{X}$, and $\mathbf{K}_{YX}$ is the covariance of $\mathbf{Y}$ and $\mathbf{X}$. Then, the marginal probability of $\mathbf{X}$ is written as

$$p(\mathbf{X}) = \int p(\mathbf{Y}, \mathbf{X})\, dX = N(\mu_X, \mathbf{K}_{XX})$$

## 9.3 Kernels

### 9.3.1 Rational Quadratic Kernel

The rational quadratic (RQ) kernel is represented by

$$k(\mathbf{y}, \mathbf{y}') = \sigma^2 \left(1 + \frac{1}{2} (\mathbf{y} - \mathbf{y}')^{\mathrm{T}} \boldsymbol{\Lambda}^{-1} (\mathbf{y} - \mathbf{y}')\right)^{-\alpha}, \text{ for } \mathbf{y}, \mathbf{y}' \in \mathbb{R}^d$$

where $\sigma^2$ is the signal variance of the function $f$ and $\boldsymbol{\Lambda} = \operatorname{diag}([\alpha_1 l_1^2, \alpha_2 l_2^2, \ldots, \alpha_d l_d^2])$ is a diagonal matrix of a product of scale mixtures $\alpha_i$, $i = 1, 2, \ldots, d$ and squared characteristic length-scales $l_i$, $i = 1, 2, \ldots, d$. A rational quadratic kernel is a generalized form of radial basis function kernel. The radial basis function of different length scales is summed up to form a rational quadratic kernel.

### 9.3.2 Exponential Sine Squared Kernel

The exponential sine squared (ESS) kernel, also called the periodic kernel, defines periodicity in prior data. The exponential sine squared kernel is given by

$$k(\mathbf{y}, \mathbf{y}') = \exp\left(\left(-2\sin^2\left(\pi (\mathbf{y} - \mathbf{y}')^{\mathrm{T}} \mathbf{p}^{-1} (\mathbf{y} - \mathbf{y}')\right)\right) \boldsymbol{\Lambda}^{-1}\right), \text{ for } \mathbf{y}, \mathbf{y}' \in \mathbb{R}^d \tag{66}$$

$\boldsymbol{\Lambda} = \operatorname{diag}([l_1^2, l_2^2, \ldots, l_d^2])$ is a diagonal matrix of squared characteristic length-scales $l_i$, $i = 1, 2, \ldots, d$. and $\mathbf{p}$ represents the periodicity.

### 9.3.3 Combined Kernels

There are two ways of combining kernels to create a new kernel: Multiplication and Addition. Multiplying kernels is the same as an AND operation because the new covariance function will have high covariance only if the covariance of the base function is increased.

The addition of kernels is equivalent to the OR operation because the new covariance function will have high covariance if one of them has high covariance, as shown in Fig. 52.

## 9.4 An Application to Flight Delay Prediction

In this section, we apply Gaussian processes to predict delays in the flights scheduled to fly in the United States national airspace. The United States Department of Transportation (DOT), and Bureau of Transportation Statistics (BTS) have broadly categorized delays reported by airlines. The most common are carrier delays, weather delays, security delays, and airspace delays. Carrier delays are attributable to the airline's management and include delays caused by maintenance, flight crew, aircraft refueling, etc. Weather delays are a consequence of extreme weather conditions which are not in the control of airlines. Airspace delays are related to the national airspace and are caused by heavy air traffic, low visibility affecting aerodrome operation, and air traffic controller delays. Security delays are caused by security concerns. Further, in the United States, the BTS also tracks the on-time performance of domestic flights operated by large air carriers and provides a summary of the number of on-time, delayed, canceled, and diverted flights that appear in DOT's monthly Air Travel Consumer Report. As per the report, 81.19% of flights in

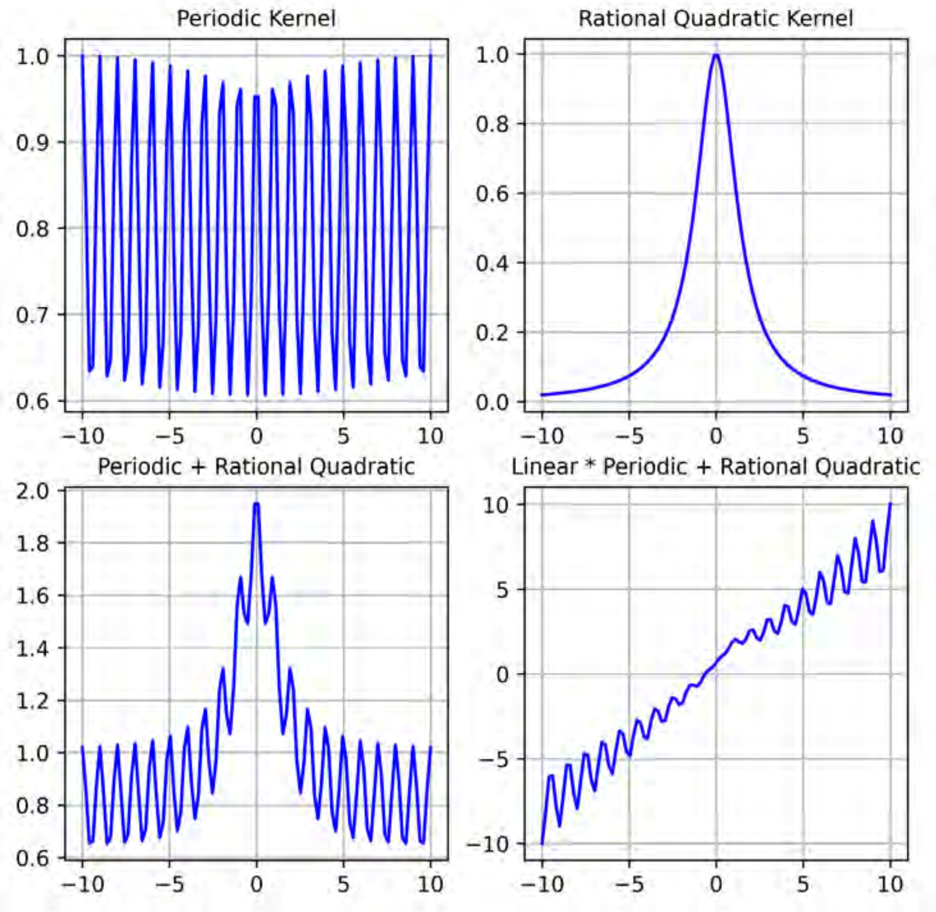


**Figure 52.—** Combined Kernel using Periodic and Rational Quadratic Kernel.

the United States were on time, and 22.31% of flights were delayed in the Dallas/Fort Worth International Airport in 2021 (Ref. 60). Thus, having prior knowledge about delays can help airlines mitigate them, if possible.

In the literature, several researchers have used various machine learning algorithms for flight delay predictions. Gui et al. (Ref. 61) used different machine learning techniques like long short-term memory (LSTM) neural networks and random forest-based models for fitting big data to predict flight delays. A prediction accuracy of 90.2 % was obtained for binary classification using a random- forest-based model. Kim et al. (Ref. 62) studied the efficacy of deep learning in forecasting air traffic delays using Recurrent Neural Networks. Similarly, Yu et al. (Ref. 63) implemented a novel deep belief network to understand patterns in flight delays. Delays can also propagate due to a chain of errors where existing flight delay further enlarges if an error happens. In flight delay, the error denotes the existence of a feature that can affect flight time. For example, technical delay in the presence of air traffic delay can further increase the flight delay time.

The objectives of this work are to identify periodicity in flight delays and predict future delays using Gaussian Process. In this section, the flight delay problem is formulated as a stochastic process dependent on a large number of random variables like departure time, weather conditions, air traffic, etc.

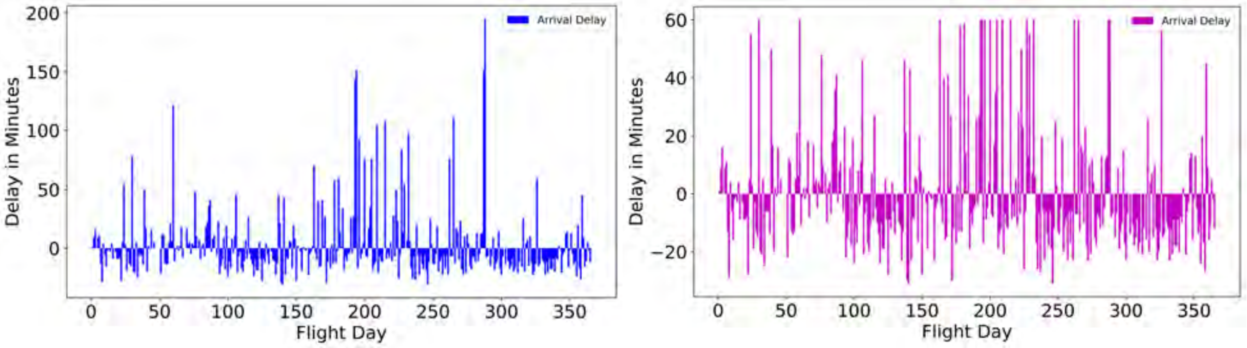
### 9.4.1 Methodology

### 9.4.2 Data Collection

In this research, flight data used are collected from two different sources. The United States DoT database is the primary source of flight information. Since the database doesn't update daily, we also used data compiled by Flightradar24, a global flight tracking service. Flightradar24 provides real-time flight information using data from Automatic Dependent Surveillance-Broadcast (ADS-B) and radar (Ref. 64). It contains flight data for all commercial flights flying in United States airspace. The available flight information from both sources includes airline flight number, departure airport, arrival airport, scheduled time of departure, actual time of departure, scheduled time of arrival, actual time of arrival, and flight time.

### 9.4.3 Data Processing

First we trim the data by removing outliers. In addition, flight diverted to nearby airports before arriving at the destination airport have longer flight times than expected. Thus, these diverted flights are not considered in the training data set. The bar plot showing the arrival delay of flight AA1567 (ATL-DFW) in 2021 is presented in Fig. 53 to a defined threshold of 60 minutes. For convenience, the flight date is converted to a numeric value by calculating the number of days since 12/30/1899 on the Gregorian calendar.



**(a)** **(b)**

**Figure 53.—** (a) Actual arrival delay of flight number AA1567 (Ref. 65) (b) Trimmed data

### 9.4.4 Training Model

We utilize *scikit-learn*, a machine learning library in Python designed for Gaussian process classification and regression (Ref. 66). The *scikit-learn* library has a large set of built-in covariance functions or kernels that can also be combined as required to create a new kernel based on the characteristics of the problem. The values of the target variable can be normalized by scaling to unit variance when zero mean and unit variance priors are used.

### 9.4.5 Flight delay Prediction

In flight delay prediction, flight departure delay and flight arrival delay are the target variables. We utilize prior flight data to select suitable kernels and their initial hyperparameters. First, we take 75 % of the prior delay time and select a kernel that correctly predicts the remaining 25 % of the delay time with a 95 % confidence interval. Then we use those kernel and hyperparameter values for the next prediction. As flight delays vary from season to season, this technique can be applied to determine a kernel that describes prior knowledge for a particular season but occasionally the selected kernel fits well for the whole year. The framework of Gaussian process regression-based prediction is shown in Fig. 54.

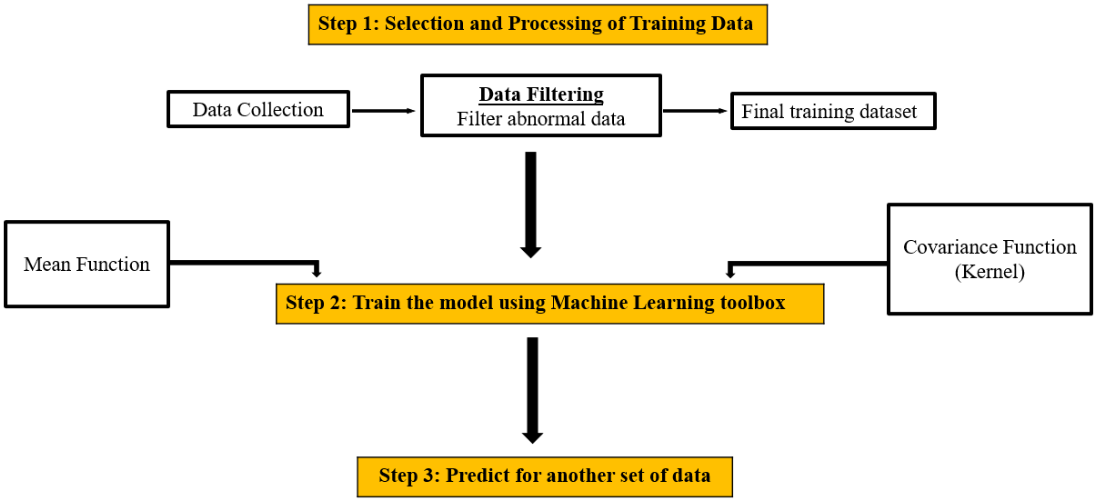

Step 1: Selection and Processing of Training Data
Data Collection
Data Filtering
Filter abnormal data
Final training dataset
Mean Function
Covariance Function (Kernel)
Step 2: Train the model using Machine Learning toolbox
Step 3: Predict for another set of data

**Figure 54.—** Framework for Gaussian Process Regression-based prediction

## 9.5 Results and Discussion

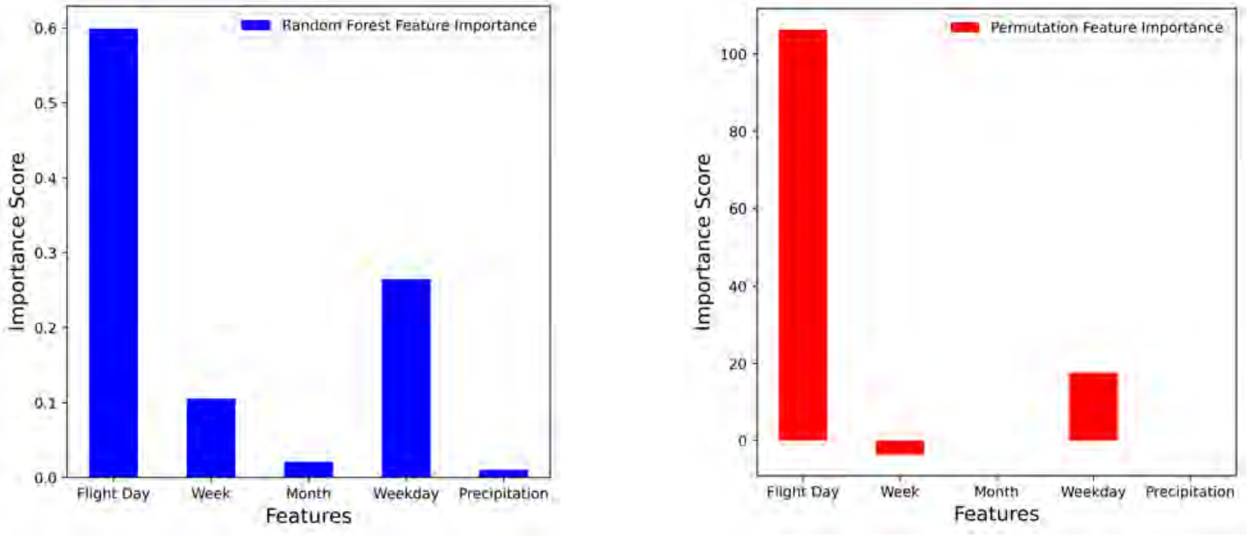
### 9.5.1 American Airlines Flight Number AA2754

American Airlines AA2754 was scheduled to fly from O'Hare International Airport (ORD), Chicago, to Dallas-Fort Worth International Airport(DFW), Texas, in 2021. First, the feature importance scores are calculated using random forest and permutation feature importance methods. Fig. 55 shows each feature's importance score. From the plot, it is obvious that the flight date is the only important feature. Finally, we trained the model with data from April 3 to July 1 and predicted the delays for the next 30 days. The kernels selected and their initial hyperparameters are given in Table. 18.

**Table 18.—** Kernels for AA2754 (ORD-DFW)

| Kernel | Hyperparameters |
|---|---|
| Constant Kernel (CK) | constant value = 1 |
| Exponential Sine Squared (ESS) | length scale ($l$) = 1, and periodicity ($p$) = 1 |
| Rational Quadratic (RQ) | length scale ($l$) = 1.5, scale mixture ($\alpha$) = 6, $l$ bounds = (1E-13,1E+5),<br>and $\alpha$ bounds = (1E-14,1E+5) |

Figure 56 compares predicted and actual delays for flight AA2754. 93% of the predicted delays were within the 95% confidence interval. The predicted delay refers to the delay in the time of arrival as predicted by the trained model. The 95% confidence interval represents the values that are two standard deviations from the average of the predicted delays. For flight AA2754, the 95% confidence interval has a range of



**(a)**

**(b)**

**Figure 55.—** AA2754 Feature Importance Score using (a) Random Forest Feature Importance (b) Permutation Feature Importance

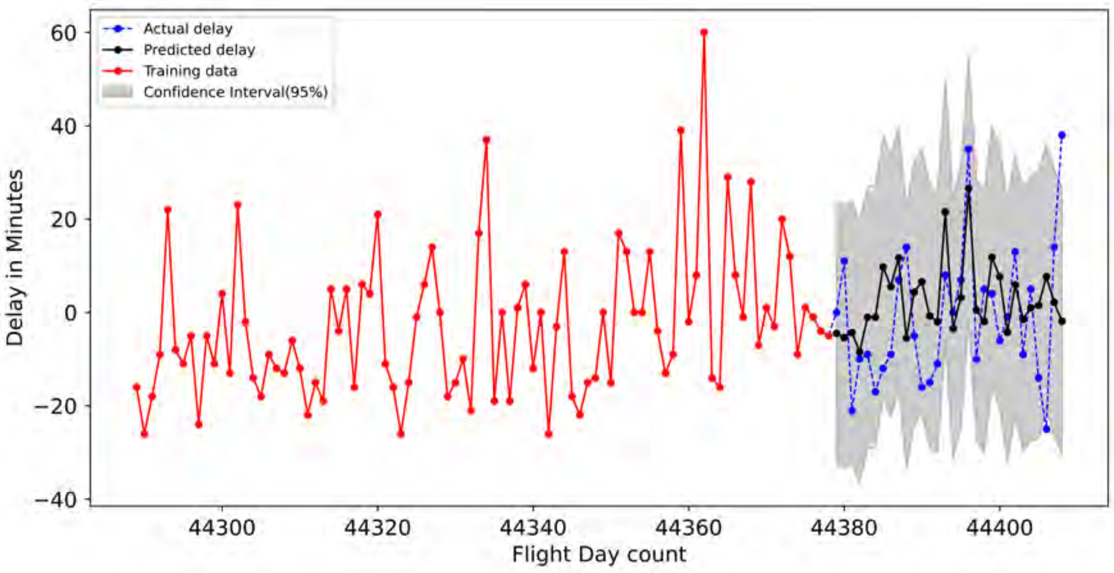
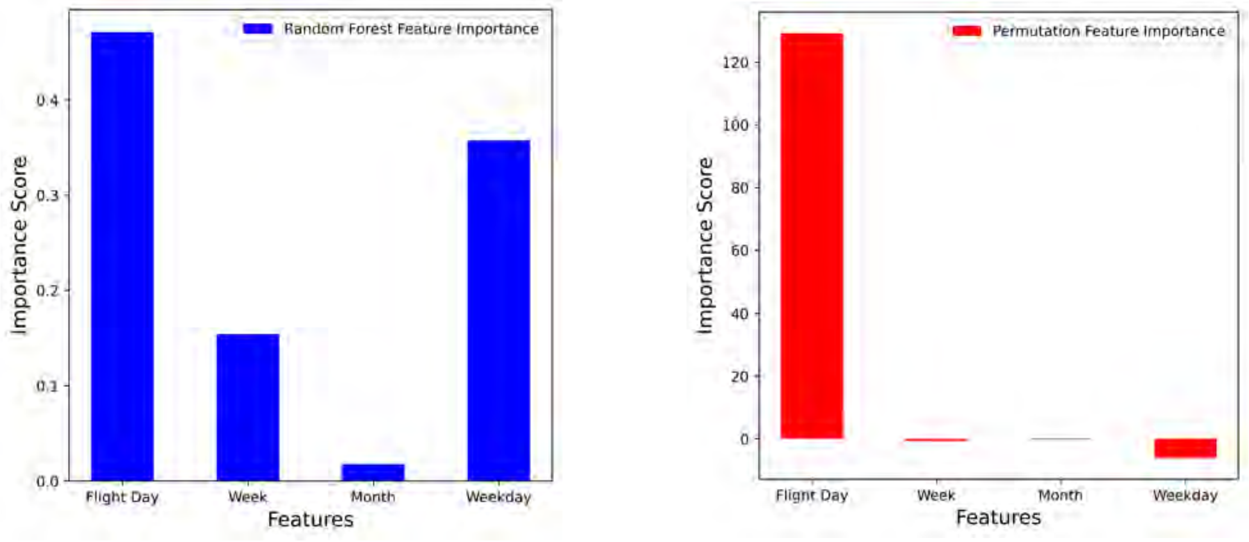


**Figure 56.—** AA2754 (ORD-DFW) predicted arrival delay versus actual arrival delay Ref. 65

approximately 60 minutes. Thus, we can predict the delay of a particular flight day within this range. For example, the predicted delay on July 19 is 26.584 ± 28 minutes with 95% confidence. The actual delay though, is 35 minutes which is well within the confidence interval, and the prediction is reasonable. Note, the actual flight delays for this time of year may differ from the predicted delays because of inconsistent delay patterns in the training data set. Further, the range of the predicted confidence interval is acceptable because flight delays are a stochastic process and can take any delay value. Thus, predicting with a narrow range of confidence interval is always a desirable feature of the algorithm.

### 9.5.2 American Airlines Flight Number AA1331 (DFW-LAX)

This flight was scheduled to fly from DFW to Los Angeles International Airport (LAX) in 2021. We trained the model with arrival delay data from 19 August 2021 to 27 September 2021 (40 flight days) and predicted the arrival delay up to 7 October 2021. We used only the flight date as a predictor variable based on the importance score shown in Fig. 57. To define the prior knowledge of the arrival delay for flight AA1331, three different kernels are combined to create a composite kernel. The kernel details are given in Table 19. The hyperparameter optimizer is set to restart 20 times.



**(a)** **(b)**

**Figure 57.—** AA1331 (DFW-LAX) Feature Importance Score using (a) Random Forest Feature Importance (b) Permutation Feature Importance

**Table 19.—** Kernels for AA1331 (DFW-LAX)

| **Kernel** | **Hyperparameters** |
|---|---|
| Constant Kernel (CK) | constant value = 1 |
| Exponential Sine Squared (ESS) | length scale ($l$) = 1, and periodicity ($p$) = 1 |
| Rational Quadratic (RQ) | length scale ($l$) = 1.5, scale mixture ($\alpha$) = 4,<br>and length scale bounds = (1E-11,1E+5) |

In Fig. 58, the actual flight delays are plotted along with predicted delays for comparison. Using only flight days as a predictor variable, for the given range of length scale bounds for the rational quadratic kernel, 100% of actual delays lie within the predicted confidence interval. However, as with the previous example, the actual flight delays in AA1331 for this time of year may differ from the predicted delays because of inconsistent delay patterns in the training data set.

Similarly, Fig. 59 shows the predicted delays when the length scale bound for the rational quadratic kernel changes to (1E-10, 1E+5). The length scale defines how smooth a function is, and as we use a hyperparameter optimizer, the prediction results with a lower length scale bound are lower from the previous

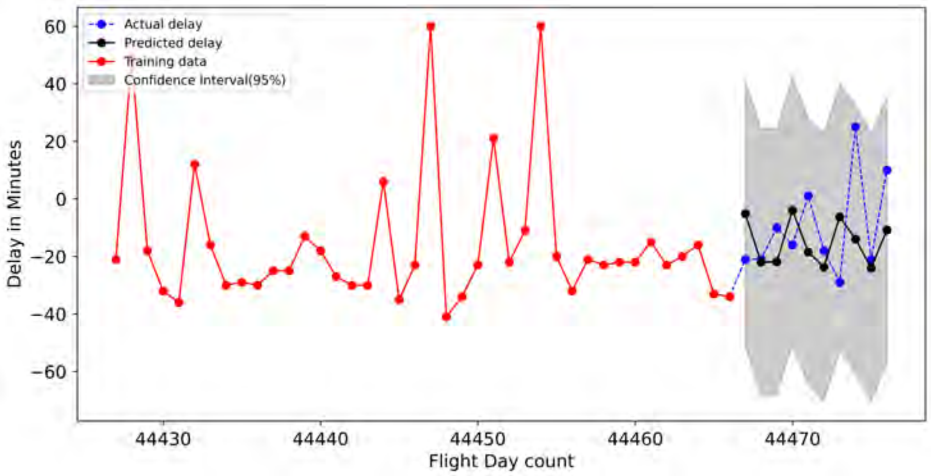


**Figure 58.—** AA1331 (DFW-LAX) predicted arrival delay versus actual arrival delay (Ref. 65)

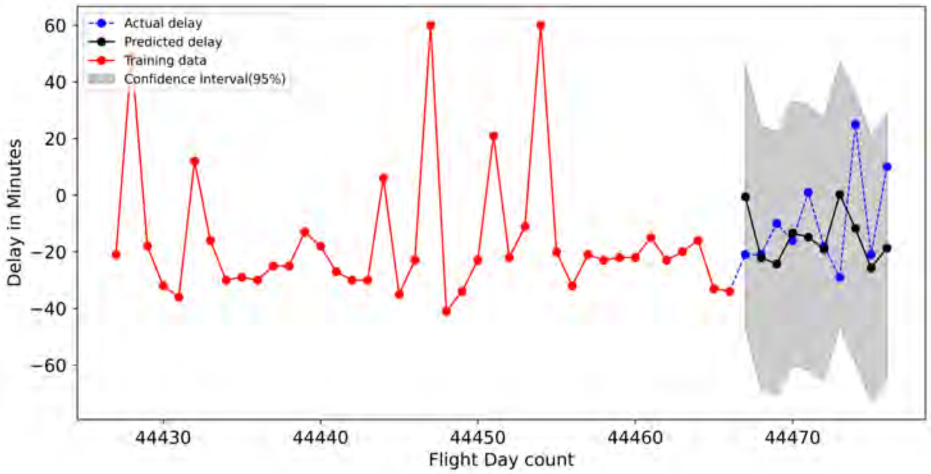
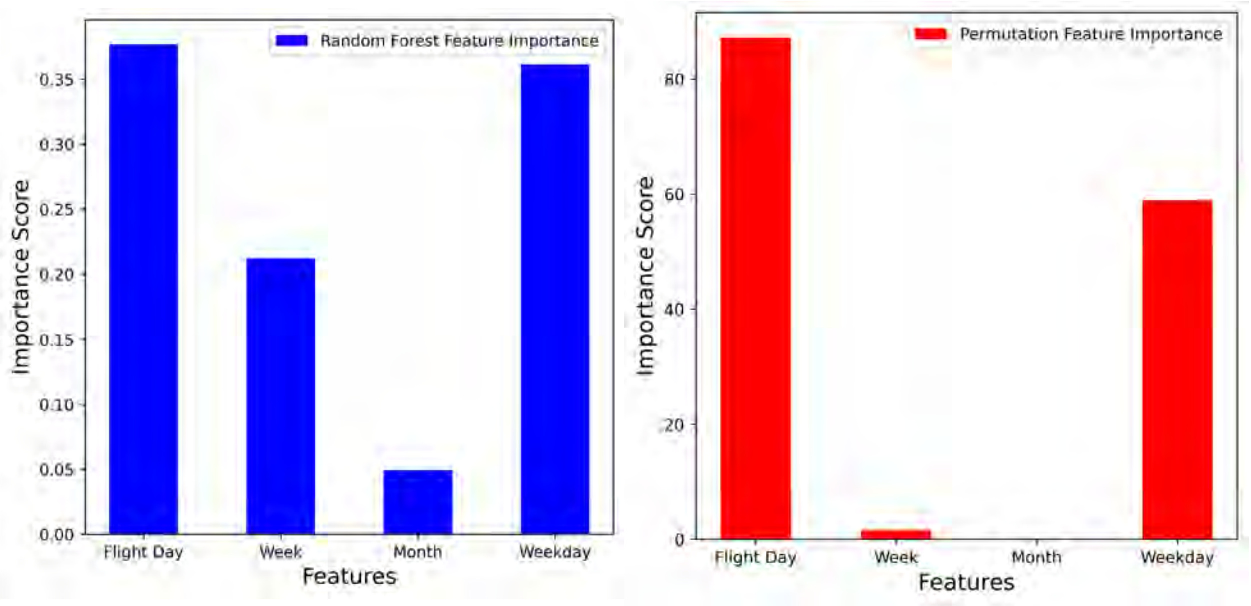


**Figure 59.—** AA1331 (DFW-LAX) predicted arrival delay versus actual arrival delay (Ref. 65) with lower length scale bound

case with a length scale bound (1E-11, 1E+5). For example, in Fig. 59, the predicted delay on October 1 is 13.38 minutes, while in Fig. 58, the predicted delay is 3.97 minutes compared to the actual delay of 5.32 minutes. Thus, we can conclude that for the dataset of flight AA1331, the hyperparameter optimizer works better with a lower length scale bound range for rational quadratic kernels.

### 9.5.3 American Airlines Flight Number AA1331 (LAX-DFW)

The same flight number, AA1331, was used by American Airlines also for a flight from LAX-DFW in 2021. Flight delay data from 18 May 2021 to 27 June 2021 is used as a training dataset to predict arrival delay for the next ten days. The kernels and hyperparameters are listed in Table. 20. Note, we identify the flight day as just an integer value referenced to the number of days since 12/30/1899 on the Gregorian calendar. The weekday is an identifier which qualifies a particular flight day. Fig. 60 shows that flight day and weekday are dominant features for prediction. However, a comparison of the importance score for the weekday calculated using both techniques depicts that flight day is more dominant over a weekday. Thus, the date of flight is the only feature used for prediction. The number of hyperparameter optimizer restarts is 20.



**(a)**

**(b)**

**Figure 60.—** AA1331 (LAX-DFW) Feature Importance Score using (a) Random Forest Feature Importance (b) Permutation Feature Importance

**Table 20.—** Kernels for AA1331 (LAX-DFW)

| **Kernel** | **Hyperparameters** |
|---|---|
| Constant Kernel (CK) | constant value = 1 |
| Exponential Sine Squared (ESS) | length scale ($l$) = 1, and periodicity ($p$) = 1 |
| Rational Quadratic (RQ) | length scale ($l$) = 1.5, scale mixture ($\alpha$) = 3,<br>and length scale bounds = (1E-9,1E+5) |

From Fig. 61, we can visualize that approximately 95% of the prediction for AA1331 (LAX-DFW) lies within the confidence interval. The 95% confidence interval has a range of 45 minutes. The predicted delay for October 3 is approximately -16 ± 22.5 minutes (i.e., the flight will arrive early), and the actual flight arrived 18 minutes earlier. The results in this case are significant only with the given length scale bounds for rational quadratic kernel because the datasets used for training the model aren't standardized. If a different length scale bound is used, the results could be different.

## 9.6 An Application to Uncertainty Propagation in an sUAV

This section focuses on the application of the Gaussian Process for Uncertainty Propagation in an sUAV. The equations of motion for a fixed-wing small UAV are given in Section 3.1

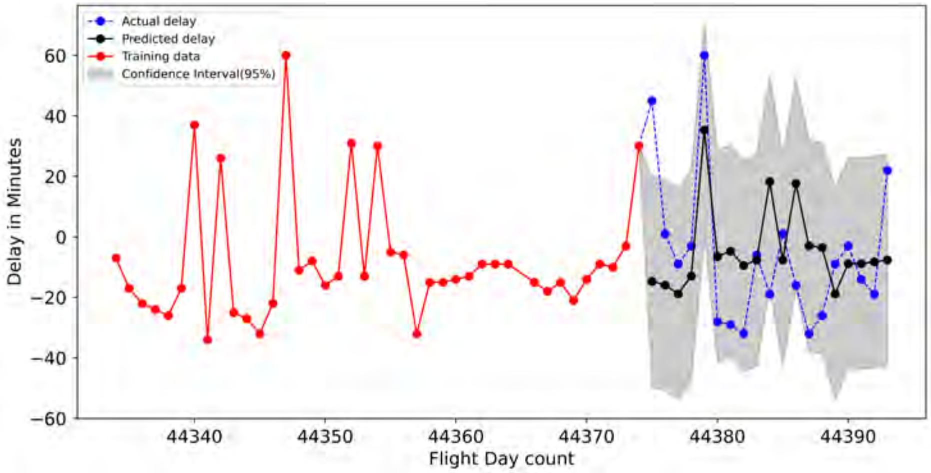


**Figure 61.—** AA1331 (LAX-DFW) predicted arrival delay versus actual arrival delay (Ref. 65)

### Case I - Randomly distributed uncertain variable

Let us consider an sUAV described as in Section 3.1 with randomly distributed maximum thrust ($T_{max}$). The parameters and their deterministic values are given in Table 21. In the simulation, time of flight is the only independent variable for predicting the trajectory of the sUAV and the maximum number of iterations for hyperparameter optimization is 25. The kernels and their hyperparameter are given in Table 22.

**Table 21.—**Uncertain variables and their pdf for Uncertainty propagation in sUAV using Gaussian Process (Case I)

| Variables | PDF/Values | Variables | PDF/Values |
|---|---|---|---|
| $x(0)$ | 0 | $H(0)$ | 250 $m$ |
| $y(0)$ | 0 | $E(0)$ | 270 $m$ |
| $C_{D0}$ | 0.0475 | $g$ | 9.81 $m/s^2$ |
| $S$ | 0.5125 $m^2$ | $\rho$ | 1.225 $Kg/m^3$ |
| $k$ | 0.02 | $\gamma_0$ | 0 $rad$ |
| $m$ | 3.125 $kg$ | $\chi(0)$ | $\pi$/3 $rad$ |
| $T_{max}$ | Random Samples [3,4]$N$ | $a_p(0)$ | $g$ |
| $\eta_c$ | 0.4 | $\lambda_n$ | 0.15 |
| $a_y(0)$ | 0.1$g$ | $\lambda_p$ | 0.15 |
| $a_{yc}$ | 0.1$g$ | $\lambda_y$ | 0.15 |
| $eta(0)$ | 1 | | |

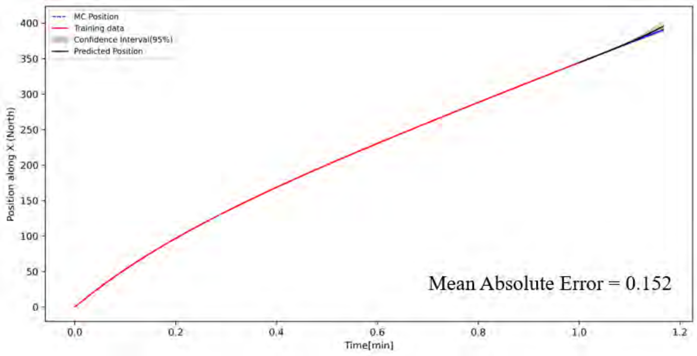
**Table 22.—** Kernels and hyperparameters for Gaussian Process (Case I)

| Kernels | Hyperparameters |
|---|---|
| Exponential Sine Squared (ESS) | Length scale ($l$) = 1, and Periodicity ($\alpha$) = 5 |
| Rational Quadratic (RQ) | Length scale ($l$) = 1, and Scale mixture ($\alpha$) = 1 |

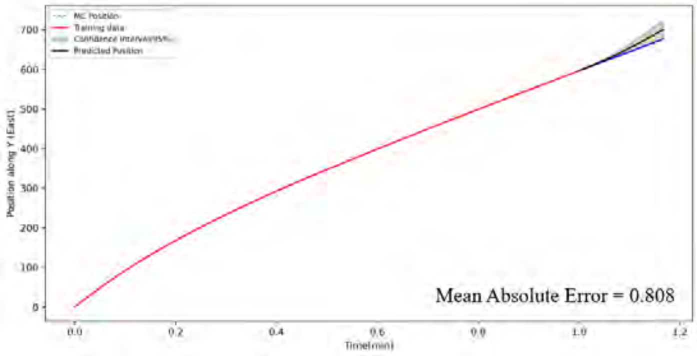
**Table 23.—** Mean absolute error($\epsilon$) meters, for prediction of time ($\Delta t$) in UAV

| **Training Data ($T_t$)** | **Mean Absolute Error ($\epsilon$) for prediction of time ($\Delta t$) along north** | | | | | |
|---|---|---|---|---|---|---|
| | $\Delta t_p$ = 5s | $\Delta t_p$ = 10s | $\Delta t_p$ = 30s | $\Delta t_p$ = 1 min | $\Delta t_p$ = 2 min | $\Delta t_p$ = 3 min |
| 30 s | 0.014 | 0.337 | 11.52 | 37.17 | 94.3 | 153.78 |
| 1 min | 0.006 | 0.152 | 6.435 | 22.21 | 57.26 | 93.074 |
| 2 min | 0.0024 | 0.061 | 3.546 | 12.15 | 30.58 | 48.27 |

| **Training Data ($T_t$)** | **Mean Absolute Error ($\epsilon$) for prediction of time ($\Delta t$) along east** | | | | | |
|---|---|---|---|---|---|---|
| | $\Delta t_p$ = 5s | $\Delta t_p$ = 10s | $\Delta t_p$ = 30s | $\Delta t_p$ = 1 min | $\Delta t_p$ = 2 min | $\Delta t_p$ = 3 min |
| 30 s | 0.3503 | 1.782 | 20.94 | 64.88 | 163.875 | 266.839 |
| 1 min | 0.1678 | 0.808 | 10.685 | 35.75 | 93.682 | 154.448 |
| 2 min | 0.1305 | 0.748 | 8.937 | 23.92 | 56.931 | 174.468 |



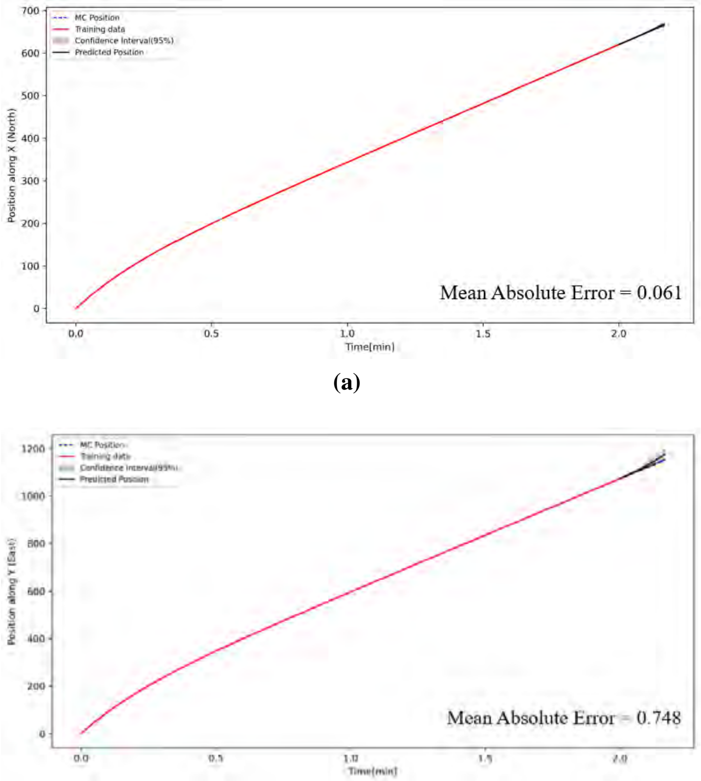
**(a)**



**(b)**

**Figure 62.—**Predicted position of UAV for 10 seconds along a) North direction (x-axis) and b) East direction (y-axis) using a model trained with preceding 1-minute data (Case I)

We first trained the model with 1 minute of preceding data and predicted the position of the sUAV for the next 10 seconds. The mean absolute errors in prediction as compared to Monte Carlo Solution is 0.152 and 0.808 meters for position along north and east respectively which is shown in Fig. 62. Next, the model is trained with 2 minutes of data. Fig. 63 illustrates that mean absolute errors reduce to 0.061 and 0.748 for positions along north and east respectively. Clearly, a larger training dataset gives a more accurate prediction. In addition, a decrease in prediction time also increases the accuracy of the prediction. Also, it



(a)

(b)

**Figure 63.—**Predicted position of sUAV for 10 seconds along a) North direction ($x$-axis) and b) East direction ($y$-axis) using a model trained with preceding 2 minutes data (Case I)

is noted that the error in position along the east is larger than the position error along the north.

### Case II - Uniformly distributed uncertain variable

In this case, we considered $C_{D0}$, $S$, $k$, $m$, and $T_{max}$ as uniformly distributed variables and Table 24 illustrates their PDF. The rest of the parameters have the same values as in the Case I discussed in Sec. 9.6. In the simulation, time of flight is the only independent variable for predicting the trajectory of the sUAV and the maximum number of iterations for hyperparameter optimization is 25. Table 25 lists the kernels and their hyperparameter.

Similar to the Case I, first, we trained the model with 1 minute of data. The mean absolute error of the predicted position as compared to the Monte Carlo Solution is 0.151 and 1.23 meters along north and east respectively. Secondly, the model is trained with 2 minutes of data, and the position of the UAV is predicted for the next 10 seconds. Fig. 65 depicts that mean absolute error decreases for both positions along north and east. The computational time for 10 seconds of prediction using a model trained with 2 minutes of data on an "Intel Core i5 4th Gen CPU @2.30 GHz" is approximately 76 seconds. The computational time was investigated because the case had more uncertain variables than Case I.

**Table 24.—**Uncertain variables and their pdf for Uncertainty propagation in sUAV using Gaussian Process (Case II)

| **Variables** | **PDF/Values** |
|---|---|
| $C_{D0}$ | $U[0.04, 0.0625]$ |
| $S$ | $U[0.41, 0.615]m^2$ |
| $k$ | $U[0.0175, 0.0225]$ |
| $m$ | $U[3, 4.25]kg$ |
| $T_{max}$ | $U[3, 4]N$ |

**Table 25.—** Kernels and hyperparameters for Gaussian Process (Case II)

| **Kernels** | **Hyperparameters** |
|---|---|
| Exponential Sine Squared (ESS) | Length scale ($l$) = 1, and Periodicity ($\alpha$) = 5 |
| Rational Quadratic (RQ) | Length scale ($l$) = 1, and Scale mixture ($\alpha$) = 1 |

**Table 26.—** Mean absolute error ($\epsilon$) meters for prediction of time ($\Delta t$) in UAV

| **Training Data ($T_t$)** | **Mean Absolute Error ($\epsilon$) for prediction of time ($\Delta t$) along north** | | | | | |
|---|---|---|---|---|---|---|
| | $\Delta t_p$ = 5s | $\Delta t_p$ = 10s | $\Delta t_p$ = 30s | $\Delta t_p$ = 1 min | $\Delta t_p$ = 2 min | $\Delta t_p$ = 3 min |
| 30 s | 0.0107 | 0.2654 | 11.56 | 38.53 | 98.69 | 161.67 |
| 1 min | 0.0062 | 0.1511 | 7.312 | 23.773 | 60.94 | 100.129 |
| 2 min | 0.001 | 0.0362 | 3.2 | 12.669 | 32.597 | 47.671 |

| **Training Data ($T_t$)** | **Mean Absolute Error ($\epsilon$) for prediction of time ($\Delta t$) along east** | | | | | |
|---|---|---|---|---|---|---|
| | $\Delta t_p$ = 5s | $\Delta t_p$ = 10s | $\Delta t_p$ = 30s | $\Delta t_p$ = 1 min | $\Delta t_p$ = 2 min | $\Delta t_p$ = 3 min |
| 30 s | 0.4056 | 2.2664 | 24.38 | 70.85 | 176.32 | 286.795 |
| 1 min | 0.2138 | 1.233 | 14.727 | 42.75 | 106.371 | 173.786 |
| 2 min | 0.102 | 0.6089 | 8.547 | 25.260 | 59.965 | 103.95 |

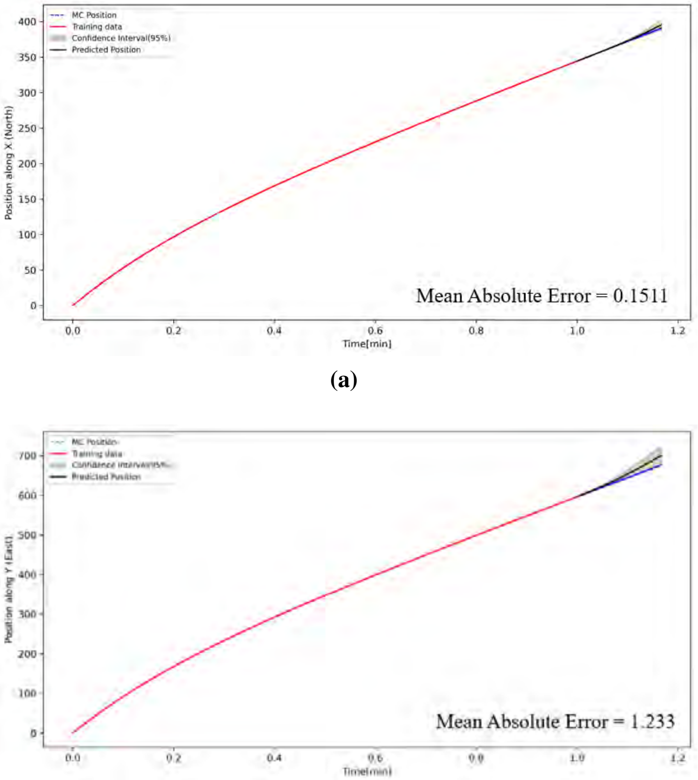
## 10 Summary and Conclusions

In this project, we studied the application of mixed sparse grid-based quadrature and generalized polynomial chaos (gPC) expansion method for uncertainty quantification and collision assessment in air traffic consisting of fixed-wing small unmanned aerial vehicles (sUAV).

We demonstrated the application of pseudospectral collocation-based gPC expansion to carry out uncertainty propagation. From the results obtained, we conclude that this provides a reliable framework to carry out quantitative conflict assessment in an unmanned air traffic, which when employed, can improve the functionalities of the unmanned traffic management system.

Further, we developed an open-source toolbox in Python for UQ and propagation in stochastic processes using the above-mentioned framework. We also demonstrated the efficacy of the toolbox with application to uncertainty quantification in prototypical systems such as the mass spring damper systems before extending the same to open loop sUAV systems.

It was observed that the results from the gPC expansion framework developed in the project can be utilized to conduct rapid probabilistic collision assessment for near real-time unmanned traffic management in the airspace. From the vehicle models, position updates, and wind-field data, apriori gPC based 3-$\sigma$



(a)

(b)

**Figure 64.—**Predicted position of sUAV for 10 seconds along a) North direction (x-axis) and b) East direction (y-axis) using a model trained with preceding 1-minute data (Case II)

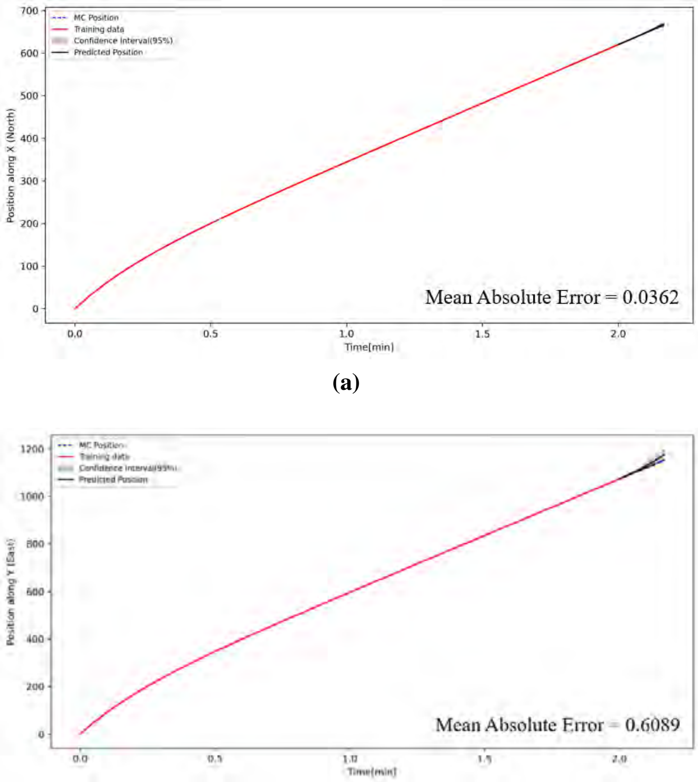
confidence ellipses can provide estimates of potential conflict at some future instants. The computational costs scaled linearly when the uncertain inputs were fewer.

Further, we calculated the largest allowable distribution of parametric uncertainties that leads to the smallest risk of collision in traffic of small unmanned aerial vehicles. The solution entailed maximizing the standard deviation for normally distributed uncertainties and maximizing the deviation of the lower and upper bounds from mean values for uniformly distributed uncertainties such that the $3-\sigma$ confidence ellipses in the position of any two vehicles in the traffic do not intersect with each other. The solution was obtained by solving a multi-objective constrained optimization.

We also established the time of closest approach between the $3-\sigma$ confidence ellipses of any two sUAVs. It was observed from numerical studies that, even small increments in the distributions of uncertainties lead to increased risk of collision in the traffic.

The separation between the sUAVs was found to be most significantly affected by uncertainties in the maximum available thrusts, zero-lift drag coefficients, and wing planform areas of the sUAVs. Moreover, studies from conflict detection in a group of heterogeneous unmanned aerial vehicles in confined airspace subject to uncertainties in the spatial gradients of the wind revealed that the mix did not pose any significant computational burden. The study of the intersection of confidence ellipses indicated that a heterogeneous mix of sUAVs resulted in an increased probability of conflict.

We also used gPC expansion-based Ensemble filtering framework in the study of conflict risk assessment in the sUAVs' traffic. It was observed that increased measurement update rate reduces the uncertainties



**(a)**

**(b)**

**Figure 65.—**Predicted position of sUAV for 10 seconds along a) North direction (x-axis) and b) East direction (y-axis) using a model trained with preceding 2 minutes data (Case II)

in the trajectories of the vehicles, further reducing the probability of conflict.

The gPC framework also provided the means to analyze vehicle impact (crash region) due to loss of control resulting from actuator failure in sUAS traffic, essentially to predict impact and crash zones for representative vehicles. The predicted regions when compared with non-participant density, provides a means to develop an early mitigation strategy, should the sUAV detect an imminent actuator failure. In such a scenario, a new sUAV path would be generated.

Finally, we studied probabilistic machine learning using Gaussian Processes with application in uncertainty quantification of stochastic processes. Results of the flight delay prediction model illustrated that Gaussian Processes Regression gives a satisfactory prediction. Further, we also implemented the machine learning technique for uncertainty propagation in sUAV. It was observed that a model trained with a larger set of training data has a smaller mean absolute error. Also, a decrease in prediction time increases the accuracy of prediction.

# References

1. Belcastro, C. M.; Newman, R. L.; Evans, J.; Klyde, D. H.; Barr, L. C.; and Ancel, E.: Hazards Identification and Analysis for Unmanned Aircraft System Operations. *17th AIAA Aviation Technology, Integration, and Operations Conference*, American Institute of Aeronautics and Astronautics, June 2017. URL `https://doi.org/10.2514/6.2017-3269`.

2. Young, S.; Ancel, E.; Moore, A.; Dill, E.; Quach, C.; Foster, J.; Darafsheh, K.; Smalling, K.; Vazquez, S.; Evans, E.; et al.: Architecture and information requirements to assess and predict flight safety risks during highly autonomous urban flight operations. , NASA, 2020.

3. Guan, X.; Lyu, R.; Shi, H.; and Chen, J.: A survey of safety separation management and collision avoidance approaches of civil UAS operating in integration national airspace system. *Chinese Journal of Aeronautics*, vol. 33, no. 11, Nov. 2020, pp. 2851–2863. URL `https://doi.org/10.1016/j.cja.2020.05.009`.

4. Pang, Y.; and Liu, Y.: Conditional Generative Adversarial Networks (CGAN) for Aircraft Trajectory Prediction considering weather effects. *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, Jan. 2020. URL `https://doi.org/10.2514/6.2020-1853`.

5. Lymperopoulos, I.; Lygeros, J.; and Lecchini, A.: Model Based Aircraft Trajectory Prediction During Takeoff. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, June 2006. URL `https://doi.org/10.2514/6.2006-6098`.

6. Sankararaman, S.; and Daigle, M.: Uncertainty Quantification in Trajectory Prediction for Aircraft Operations. *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, Jan. 2017. URL `https://doi.org/10.2514/6.2017-1724`.

7. Cook, A.; Blom, H. A.; Lillo, F.; Mantegna, R. N.; Miccichè, S.; Rivas, D.; Vázquez, R.; and Zanin, M.: Applying complexity science to air traffic management. *Journal of Air Transport Management*, vol. 42, Jan. 2015, pp. 149–158. URL `https://doi.org/10.1016/j.jairtraman.2014.09.011`.

8. Roychoudhury, I.; Spirkovska, L.; Daigle, M.; Balaban, E.; Sankararaman, S.; Kulkarni, C. S.; Poll, S.; and Goebel, K.: Predicting Real-Time Safety of the National Airspace System. *AIAA Infotech @ Aerospace*, American Institute of Aeronautics and Astronautics, Jan. 2016. URL `https://doi.org/10.2514/6.2016-2131`.

9. Ribeiro, M.; Ellerbroek, J.; and Hoekstra, J.: Review of Conflict Resolution Methods for Manned and Unmanned Aviation. *Aerospace*, vol. 7, no. 6, June 2020, p. 79. URL `https://doi.org/10.3390/aerospace7060079`.

10. Lymperopoulos, I.; and Lygeros, J.: Sequential Monte Carlo methods for multi-aircraft trajectory prediction in air traffic management. *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, Apr. 2010, pp. 830–849. URL `https://doi.org/10.1002/acs.1174`.

11. Fishman, G.: *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media, 2013.

12. Xiu, D.; and Hesthaven, J. S.: High-Order Collocation Methods for Differential Equations with Random Inputs. *SIAM Journal on Scientific Computing*, vol. 27, no. 3, Jan. 2005, pp. 1118–1139. URL `https://doi.org/10.1137/040615201`.

13. Bhusal, R.; and Subbarao, K.: Uncertainty Quantification Using Generalized Polynomial Chaos Expansion for Nonlinear Dynamical Systems With Mixed State and Parameter Uncertainties. *ASME Journal of Computational and Nonlinear Dynamics*, vol. 14, no. 2, Jan. 2019. URL `https://doi.org/10.1115/1.4041473`.

14. Prabhakar, A.; Fisher, J.; and Bhattacharya, R.: Polynomial Chaos-Based Analysis of Probabilistic Uncertainty in Hypersonic Flight Dynamics. *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, Jan. 2010, pp. 222–234. URL `https://doi.org/10.2514/1.41551`.

15. Hosder, S.; and Walters, R.: Non-Intrusive Polynomial Chaos Methods for Uncertainty Quantification in Fluid Dynamics. *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Jan. 2010. URL `https://doi.org/10.2514/6.2010-129`.

16. Casado, E.; Civita, M. L.; Vilaplana, M.; and McGookin, E. W.: Quantification of aircraft trajectory prediction uncertainty using polynomial chaos expansions. *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, IEEE, Sept. 2017. URL `https://doi.org/10.1109/dasc.2017.8102052`.

17. Xiu, D.; and Karniadakis, G. E.: The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM Journal on Scientific Computing*, vol. 24, no. 2, Jan. 2002, pp. 619–644. URL `https://doi.org/10.1137/s1064827501387826`.

18. Jones, B. A.; Parrish, N.; and Doostan, A.: Postmaneuver Collision Probability Estimation Using Sparse Polynomial Chaos Expansions. *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 8, Aug. 2015, pp. 1425–1437. URL `https://doi.org/10.2514/1.g000595`.

19. Bhusal, R.; and Subbarao, K.: Sensitivity Analysis of Cooperating Multi-agent Systems with Uncertain Connection Weights. *2019 American Control Conference (ACC)*, IEEE, July 2019. URL `https://doi.org/10.23919/acc.2019.8815336`.

20. McConaghy, T.; Breen, K.; Dyck, J.; and Gupta, A.: 3-Sigma Verification and Design. *Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide*, Springer New York, Sept. 2012, pp. 65–114. URL `https://doi.org/10.1007/978-1-4614-2269-3_4`.

21. Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, vol. 93, no. 7, July 2008, pp. 964–979. URL `https://doi.org/10.1016/j.ress.2007.04.002`.

22. Jenie, Y. I.; van Kampen, E.-J.; Ellerbroek, J.; and Hoekstra, J. M.: Safety Assessment of a UAV CD&R System in High Density Airspace Using Monte Carlo Simulations. *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, Aug. 2018, pp. 2686–2695. URL `https://doi.org/10.1109/tits.2017.2758859`.

23. Hernández-Romero, E.; Valenzuela, A.; and Rivas, D.: Probabilistic multi-aircraft conflict detection and resolution considering wind forecast uncertainty. *Aerospace Science and Technology*, vol. 105, Oct. 2020, p. 105973. URL `https://doi.org/10.1016/j.ast.2020.105973`.

24. Zheng, Q.; and Zhao, Y.: Modeling Wind Uncertainties for Stochastic Trajectory Synthesis. *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, American Institute of Aeronautics and Astronautics, June 2011. URL `https://doi.org/10.2514/6.2011-6858`.

25. Chaloulos, G.; and Lygeros, J.: Effect of Wind Correlation on Aircraft Conflict Probability. *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, Nov. 2007, pp. 1742–1752. URL `https://doi.org/10.2514/1.28858`.

26. Hernández-Romero, E.; Valenzuela, A.; and Rivas, D.: A probabilistic approach to measure aircraft conflict severity considering wind forecast uncertainty. *Aerospace Science and Technology*, vol. 86, Mar. 2019, pp. 401–414. URL `https://doi.org/10.1016/j.ast.2019.01.024`.

27. Matsuno, Y.; Tsuchiya, T.; Wei, J.; Hwang, I.; and Matayoshi, N.: Stochastic optimal control for aircraft conflict resolution under wind uncertainty. *Aerospace Science and Technology*, vol. 43, June 2015, pp. 77–88. URL `https://doi.org/10.1016/j.ast.2015.02.018`.

28. Gilitschenski, I.; and Hanebeck, U. D.: A direct method for checking overlap of two hyperellipsoids. *2014 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, IEEE, Oct. 2014. URL `https://doi.org/10.1109/sdf.2014.6954724`.

29. Chihara, T. S.: The Three Term Recurrence Relation and Spectral Properties of Orthogonal Polynomials. *Orthogonal Polynomials*, Springer Netherlands, 1990, pp. 99–114. URL `https://doi.org/10.1007/978-94-009-0501-6_4`.

30. Gubner, J. A.: *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006. URL `https://doi.org/10.1017/cbo9780511813610`.

31. Xiu, D.: *Numerical methods for stochastic computations: a spectral method approach*. Princeton University Press, 2010. ISBN: 978-0-691-14212-8.

32. Askey, R.; and Wilson, J.: Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials. *Memoirs of the American Mathematical Society*, vol. 54, no. 319, 1985. URL `https://doi.org/10.1090/memo/0319`.

33. Xiu, D.; and Karniadakis, G. E.: The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM Journal on Scientific Computing*, vol. 24, no. 2, Jan. 2002, pp. 619–644. URL `https://doi.org/10.1137/s1064827501387826`.

34. Jia, B.; Xin, M.; and Cheng, Y.: Sparse Gauss-Hermite Quadrature Filter with Application to Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, Mar. 2011, pp. 367–379. URL `https://doi.org/10.2514/1.52016`.

35. Shippey, B.; and Subbarao, K.: Trajectory design using collocation and genetic algorithms: Aircraft turning maneuver. *2008 IEEE International Conference on Control Applications*, IEEE, 2008. URL `https://doi.org/10.1109/cca.2008.4629636`.

36. Segal, S.; Ben-Asher, J. Z.; and Weiss, H.: Derivation of Formation-Flight Guidance Laws for Unmanned Air Vehicles. *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, July 2005, pp. 733–742. URL https://doi.org/10.2514/1.7420.

37. Miele, A.; Wang, T.; and Melvin, W. W.: Optimization and acceleration guidance of flight trajectories in a windshear. *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, July 1987, pp. 368–377. URL https://doi.org/10.2514/3.20227.

38. Bhusal, R.; and Subbarao, K.: Generalized Polynomial Chaos-based Ensemble Kalman Filtering for Orbit Estimation. *2021 American Control Conference (ACC)*, IEEE, May 2021. URL https://doi.org/10.23919/acc50511.2021.9482961.

39. Bhusal, R.: Uncertainty Propagation, Control, and Estimation of Stochastic Dynamic Systems Using Generalized Polynomial Chaos Expansion. Ph.D. Thesis, The University of Texas at Arlington, 2021.

40. Bhusal, R.; and Subbarao, K.: Generalized Polynomial Chaos Expansion Approach for Uncertainty Quantification in Small Satellite Orbital Debris Problems. *The Journal of the Astronautical Sciences*, vol. 67, no. 1, May 2019, pp. 225–253. URL https://doi.org/10.1007/s40295-019-00176-1.

41. Tang, Y.; Kurths, J.; Lin, W.; Ott, E.; and Kocarev, L.: Introduction to Focus Issue: When machine learning meets complex systems: Networks, chaos, and nonlinear dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 6, June 2020, p. 063151. URL https://doi.org/10.1063/5.0016505.

42. Tanevski, J.; Todorovski, L.; and Džeroski, S.: Learning stochastic process-based models of dynamical systems from knowledge and data. *BMC systems biology*, vol. 10, no. 1, 2016, pp. 1–17.

43. Jiang, L.; Wang, L.; and Zhou, K.: Deep learning stochastic processes with QCD phase transition. *Physical Review D*, vol. 103, no. 11, 2021, p. 116023.

44. Cao, G.; Lai, E. M.-K.; and Alam, F.: Gaussian process model predictive control of an unmanned quadrotor. *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, 2017, pp. 147–162.

45. Palar, P. S.; Zakaria, K.; Zuhal, L. R.; Shimoyama, K.; and Liem, R. P.: Gaussian Processes and Support Vector Regression for Uncertainty Quantification in Aerodynamics. *AIAA Scitech 2021 Forum*, 2021, p. 0181.

46. Brahim-Belhouari, S.; and Bermak, A.: Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, vol. 47, no. 4, 2004, pp. 705–712.

47. Idé, T.; and Kato, S.: Travel-time prediction using Gaussian process regression: A trajectory-based approach. *Proceedings of the 2009 SIAM International Conference on Data Mining*, SIAM, 2009, pp. 1185–1196.

48. Amer, A. W.; Roy, S.; and Kopsaftopoulos, F.: Probabilistic SHM under varying loads via the integration of Gaussian Process Regression and physics-based guided-wave propagation models. *AIAA Scitech 2021 Forum*, 2021, p. 0434.

49. Mohanty, S.; Chattopadhyay, A.; Peralta, P.; Das, S.; and Willhauck, C.: Fatigue life prediction using multivariate Gaussian process. *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference, 10th AIAA Non-Deterministic Approaches Conference, 9th AIAA Gossamer Spacecraft Forum, 4th AIAA Multidisciplinary Design Optimization Specialists Conference*, 2008, p. 1837.

50. Lee, H.; Malik, W.; Zhang, B.; Nagarajan, B.; and Jung, Y. C.: Taxi time prediction at Charlotte Airport using fast-time simulation and machine learning techniques. *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015, p. 2272.

51. Ross, S.: *A first course in probability*, Pearson, New Jersey, 5. 8 ed., 2010, p. 198.

52. Liu, M.; Chowdhary, G.; da Silva, B. C.; Liu, S.-Y.; and How, J. P.: Gaussian Processes for Learning and Control: A Tutorial with Examples. *IEEE Control Systems*, vol. 38, no. 5, Oct. 2018, pp. 53–86. URL `https://doi.org/10.1109/mcs.2018.2851010`.

53. Duvenaud, D.: Automatic model construction with Gaussian processes. Ph.D. Thesis, University of Cambridge, 2014.

54. Williams, C. K.; and Rasmussen, C. E.: *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.

55. Wang, J.: An intuitive tutorial to Gaussian processes regression. *arXiv preprint arXiv:2009.10862*, 2020.

56. Schulz, E.; Speekenbrink, M.; and Krause, A.: A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, vol. 85, 2018, pp. 1–16.

57. Ebden, M.: Gaussian processes: A quick introduction. *arXiv preprint arXiv:1505.02965*, 2015.

58. Murphy, K. P.: *Machine learning: a probabilistic perspective*. MIT press, 2012.

59. Wakefield, J.: *Bayesian and frequentist regression methods*, vol. 23. Springer, 2013.

60. United States Department of Transportation Bureau of Transportation Statistics,On-Time Arrival Performance Dallas/Fort Worth, TX: Dallas/Fort Worth International (January - December, 2021)[Data File]. 2022. URL `https://www.transtats.bts.gov/ot_delay/OT_DelayCause1.asp`.

61. Gui, G.; Liu, F.; Sun, J.; Yang, J.; Zhou, Z.; and Zhao, D.: Flight delay prediction based on aviation big data and machine learning. *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, 2019, pp. 140–150.

62. Kim, Y. J.; Choi, S.; Briceno, S.; and Mavris, D.: A deep learning approach to flight delay prediction. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, IEEE, 2016, pp. 1–6.

63. Yu, B.; Guo, Z.; Asian, S.; Wang, H.; and Chen, G.: Flight delay prediction for commercial air transport: A deep learning approach. *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, 2019, pp. 203–221.

64. Live flight tracker - real-time flight tracker map. 2022. URL `https://www.flightradar24.com/how-it-works`.

65. Detailed Statistics Arrivals,United States Department of Transportation Bureau of Transportation Statistics. `https://www.transtats.bts.gov/ontime/Arrivals.aspx` [cited 15 May 2022], 2022.

66. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825–2830.

67. Petzke, F.; Mesbah, A.; and Streif, S.: PoCET: a Polynomial Chaos Expansion Toolbox for Matlab. *IFAC-PapersOnLine*, vol. 53, no. 2, 2020, pp. 7256–7261.

68. Oladyshkin, S.; and Nowak, W.: Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, vol. 106, Oct. 2012, pp. 179–190.

69. Miller, S.; and Childers, D.: *Probability and random processes: With applications to signal processing and communications*. Academic Press, 2012.

# Appendix A

# Python Toolbox for Generalized Polynomial Chaos Expansion-based Uncertainty Quantification

There are various toolboxes available for uncertainty quantification using gPC expansion. Petzke et al. Ref. 67 developed a MATLAB toolbox (PoCET). This toolbox supports the calculation of the coefficient of expansion and the second moment of a stochastic process. Oladyshkin et al. Ref. 68 proposed arbitrary polynomial chaos (aPC) for uncertainty quantification which uses arbitrary distribution and probability density function. However, only a few of these toolboxes are open-source. In addition, existing toolboxes aren't useful for stochastic processes with the joint probability distribution of uncertainties with different marginal density types (e.g., problems where some of the uncertainties are uniformly distributed while others are normally distributed).

In this section, we develop a Python toolbox for uncertainty quantification and propagation in stochastic processes using generalized polynomial chaos expansion. We utilized Python for development because it is a very easily accessible programming language, codes are easy to understand, and Python provides comprehensive libraries for visualizing results. The developed toolbox uses Numpy and Scipy libraries to handle arrays and stochastic differential equations (SDEs). In addition, a critical aspect of this toolbox is the mixed sparse grid quadrature technique, which can give collocation points for SDEs having random variables with different probability distributions i.e., Gaussian and Uniform distribution.

## Toolbox Description

In this section, we provide structure and detailed functional descriptions of prime modules in the toolbox. The reader can refer appendix for detail on the toolbox repository.

### Main module

This toolbox requires the user to create a main module as required. The main module can be divided into three sections: Pre-processor, Processor/Solver, and Post-processor. This section covers details on the structure of the main module.

### Pre-processor

In the pre-processor section, we need to define the model using *Parameter class* followed by *Uncertainty Class* and *MSG Class*. Attributes of *Parameter Class* are all the variables including initial conditions required to solve SDEs. However, initial conditions and parameters, if considered as random variables should be characterized by probability distribution function and are defined inside *Uncertainty Class*. For example, in a mass spring damper system with uncertain initial conditions, represented by $m\ddot{x} + c\dot{x} + kx = 0$, the initial condition of position $x(0)$ and velocity $\dot{x}(0)$ are attributes of *Uncertainty Class* and parameters $m$, $c$, and $k$ are attributes of *Parameter Class*. *MSG Class* defines attributes of MSG quadrature using four variables to create collocation nodes and node weights. The four variables are the dimension of uncertain variables ($d$), number of Gaussian random variables ($d\_Gaussian$), number of uniform random variables ($d\_Uniform$), and maximum accuracy level ($L$). This toolbox is designed to support two types of probability distribution: Uniform and Gaussian. Importantly, most of the physical systems can be represented as a stochastic process using Gaussian Ref. 69 and uniform random variables. The reader can refer to Section A for illustrative code.

**Processor/Solver**

Processor/Solver section includes developing gPC sample nodes and respective node weights, evaluating gPC basis and finally solving for deterministic cases. Mixed sparse grid techniques are implemented to develop sample nodes and node weights.

**Post processor**

In post-processing, we will calculate gPC coefficients, mean, covariance, and any other higher-order moments of the probability density function as needed.

### MSG Modules

The *MSG modules* are a part of the proposed toolbox. There are altogether 22 modules for the mixed sparse grid quadrature technique. These modules are interrelated and the user is only required to define four variables to get the gPC sample node and node weights. The four variables are the dimension of uncertain variables ($d$), number of Gaussian random variables ($d\_Gaussian$), number of uniform random variables ($d\_Uniform$), and maximum accuracy level ($L$). As discussed in the main module subsection, these variables are defined as attributes of *MSG class*. This module uses nested Kronrod Patterson Hermite and Patterson Legendre univariate rules for computing collocation nodes.

### gPC Module

gPC modules are Python files coded for a particular assignment in gPC expansion. These modules are imported into the main module as well as other gPC modules. The proposed toolbox has thirteen modules specifically for gPC expansion. Each module has a function of the same name defined inside it. In this section, we will discuss in detail the functions of each module.

**gpc_ essentials module**

*gpc_ essentials module* has inbuilt *gpc_ essentials function* which returns variable *syschar*. Variable *syschar* is concatenated string denoting the number of orthogonal polynomials for an underlying random variable. In concatenated string, the Hermite polynomial is represented by "$H$" and the Legendre polynomial is represented by "$L$". If the stochastic system has two Gaussian random variables and one uniform random variable, the variable *syschar* = "$HHL$". The sequence of concatenation is Hermite polynomial followed by Legendre Polynomial.

**gpcbasis_ evaluate module**

*gpcbasis_ evaluate function* takes *syschar*, *indexes* and nodes from MSG quadrature as input arguments. This module evaluates basis functions i.e., Hermite and Legendre polynomials at nodes created by *MSG module*. It imports *hermite_ poly* and *legendre_ shifted_ poly* modules to perform basis evaluation based on probability distribution.

**gpc_ coefficient_ cal module**

This module computes the gPC expansion coefficient for propagation solution at collocation nodes. *gpc_ coefficient_ cal function* takes gPC normalization coefficient, gPC basis array calculated at nodes, node weights, and solution at nodes as arguments.

**gaussian_transformation module**

This module can transform normally distributed standard quadrature nodes to normally distributed collocation nodes having a mean and standard deviation of given normally distributed uncertain variables.

**gpc_ covariance module**

*gpc_ covariance function* is coded to determine covariance between two stochastic processes. This module requires the gPC normalization coefficient and gPC coefficient calculated by *gpc_ coefficient_ cal function* as inputs. It will return the covariance matrix of two random processes.

**hermite_ poly module**

This module computes the Hermite polynomial of degree $n$ at a particular node value which is passed as an argument. Elements of the indexes matrix are also passed as an argument to *hermite_poly function*.

**legendre_ shifted_ poly module**

*legendre_ shifted_ poly module* computes Legendre polynomial of degree $n$ at particular node. The input arguments to *legendre_ shifted_ poly function* are the same as arguments passed to *hermite_ poly function*. However, only one function is called at a time because the call depends on probability distribution.

**Indexes module**

*Indexes function* inside this module is necessary for *multi_ index module* to work. This function is called by *multi_ index module* passing polynomial chaos order ($P$) and several uncertain variables ($d$) as arguments. This module is designed to create a vector of indexes for the degrees of the different orthogonal polynomials where P constraints maximum value inside the indexes.*Indexes module* returns gPC order ($N$) and vector of indexes. gPC order ($N$) is binomial coefficient of ($P$+$d$) and $d$.

**multi_ index module**

This module is designed to return the normalization coefficient and order of gPC expansion. Polynomial chaos order ($P$), number of uncertain variables ($d$), and a string *syschar* are passed as arguments to *multi_ index function*. The normalization coefficient for each orthogonal polynomial is calculated using the respective formula i.e., $n!$ for the Hermite polynomial and $\frac{2n}{2n+1}$ for the Legendre polynomial. The normalization coefficient for a multi-index system is the product of each normalization coefficient.

**uniform_transformation module**

*uniform_transformation module* transforms uniformly distributed quadrature nodes having PDF $U[0, 1]$ to uniformly distributed collocation nodes with PDF ranging from lower range to upper range defined by the user. Figure. A1 shows code snippet of *uniform_transformation module*.

### ODE Module

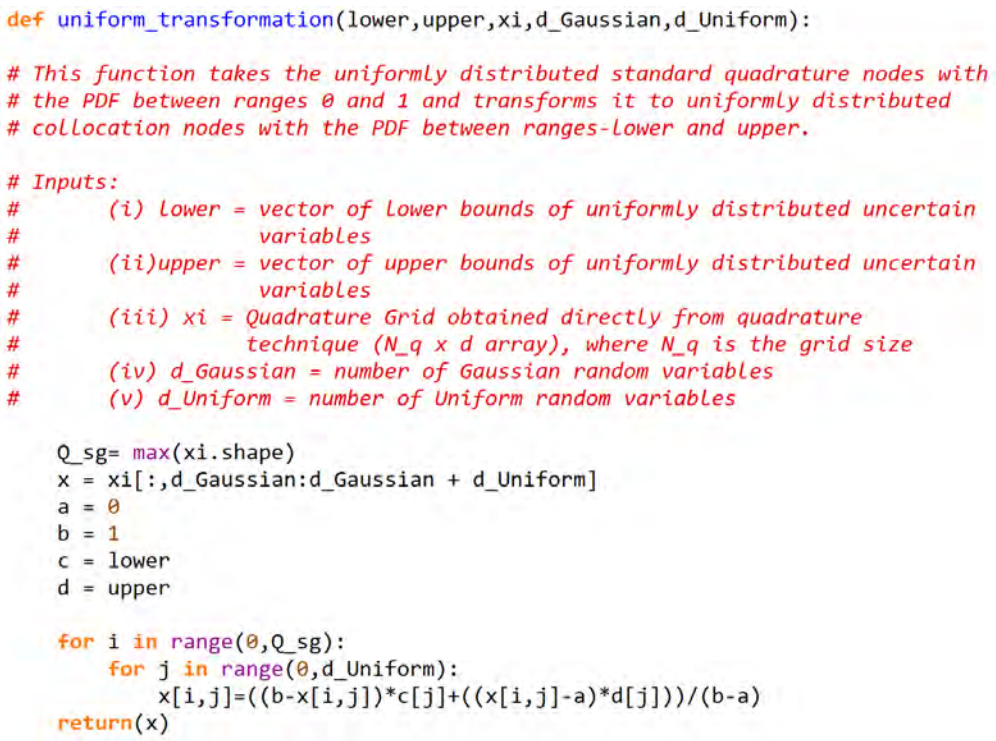
It is the user's responsibility to develop *ode module* with *ode function* defined inside it. Ordinary differential equation solver available in *Scipy library* requires higher order ODE to be converted to first order ODE. An algorithm for *ode function* for a mass spring damper system is given in this section. Figure. A2 shows code snippet of *ode module* in Python.

```
function ODE(t,x,m,c,k)                ▷ t ← time vector, x ← initial conditions, m,c,k ← parameters
    $x1\_dot$ ← x[1]                                                       ▷ Position
    $x2\_dot$ ← - (c/m)* x[1] - (k/m)* x[0]                                ▷ Velocity
    return $x1\_dot$ $x2\_dot$
end function
```

**Algorithm 1:** ODE Module

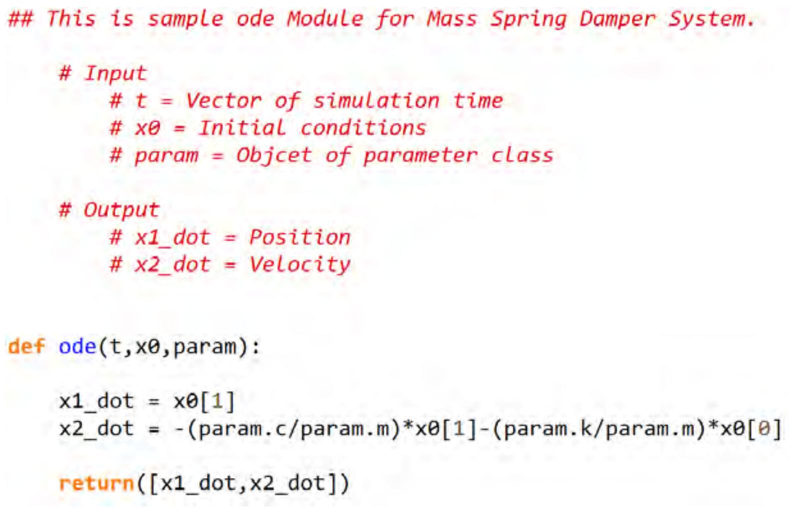
## Illustrative Example

In this section, we verify the functionality of the toolbox using uncertainty quantification examples in a mass spring damper system. We test the proposed toolbox for two different stochastic processes. The



```python
def uniform_transformation(lower,upper,xi,d_Gaussian,d_Uniform):

# This function takes the uniformly distributed standard quadrature nodes with
# the PDF between ranges 0 and 1 and transforms it to uniformly distributed
# collocation nodes with the PDF between ranges-lower and upper.

# Inputs:
#       (i) lower = vector of lower bounds of uniformly distributed uncertain
#                   variables
#       (ii)upper = vector of upper bounds of uniformly distributed uncertain
#                   variables
#       (iii) xi = Quadrature Grid obtained directly from quadrature
#                  technique (N_q x d array), where N_q is the grid size
#       (iv) d_Gaussian = number of Gaussian random variables
#       (v) d_Uniform = number of Uniform random variables

    Q_sg= max(xi.shape)
    x = xi[:,d_Gaussian:d_Gaussian + d_Uniform]
    a = 0
    b = 1
    c = lower
    d = upper

    for i in range(0,Q_sg):
        for j in range(0,d_Uniform):
            x[i,j]=((b-x[i,j])*c[j]+((x[i,j]-a)*d[j]))/(b-a)
    return(x)
```

**Figure A1.—** Code snippet of $uniform\_transformation$ module in proposed toolbox



```python
## This is sample ode Module for Mass Spring Damper System.

    # Input
        # t = Vector of simulation time
        # x0 = Initial conditions
        # param = Objcet of parameter class

    # Output
        # x1_dot = Position
        # x2_dot = Velocity


def ode(t,x0,param):

    x1_dot = x0[1]
    x2_dot = -(param.c/param.m)*x0[1]-(param.k/param.m)*x0[0]

    return([x1_dot,x2_dot])
```

**Figure A2.—**Code snippet of $ode$ module for mass spring damper

solutions from the Python toolbox are compared with solutions from Monte Carlo Simulation. In addition, a comparison of computational cost for gPC and Monte Carlo simulation is presented.

### Case I - Uncertain Initial Condition

Let us consider an unforced mass spring damper system represented by $m\ddot{x} + c\dot{x} + kx = 0$, where $m$ is mass, $c$ is damping coefficient, and $k$ is spring constant. We will first perform uncertainty quantification and then propagate it over the system. Let us assume that the initial condition of position $x(0)$ and velocity $\dot{x}(0)$ are random variables and have a Gaussian probability density function, as shown in Table A1. The deterministic value of parameters used for simulation is given in Table A2.

**Table A1.—**Uncertain variables and their PDF (Case I)

| Variable | Distribution | PDF |
|---|---|---|
| $x(0)$ | Normal | $N$ (4,$1^2$) $m$ |
| $\dot{x}(0)$ | Normal | $N$ (1,$0.1^2$) $m/s$ |

**Table A2.—**Deterministic parameters (Case I)

| Parameters | Values |
|---|---|
| Mass ($m$) | 4 $Kg$ |
| Damping Coefficient ($c$) | 2 $Ns/m$ |
| Spring Constant ($k$) | 45 $N/m$ |

Let us assume the polynomial chaos order ($P$) = 3 and the accuracy level for MSG quadrature ($L$) = 6. The polynomial chaos order can be changed to higher order as required depending on the convergence of the solution. The solution for position and velocity using the Python toolbox is shown in Fig. A3. Figure A4 shows the solution of the stochastic process with the uncertain initial condition using Monte Carlo simulation.

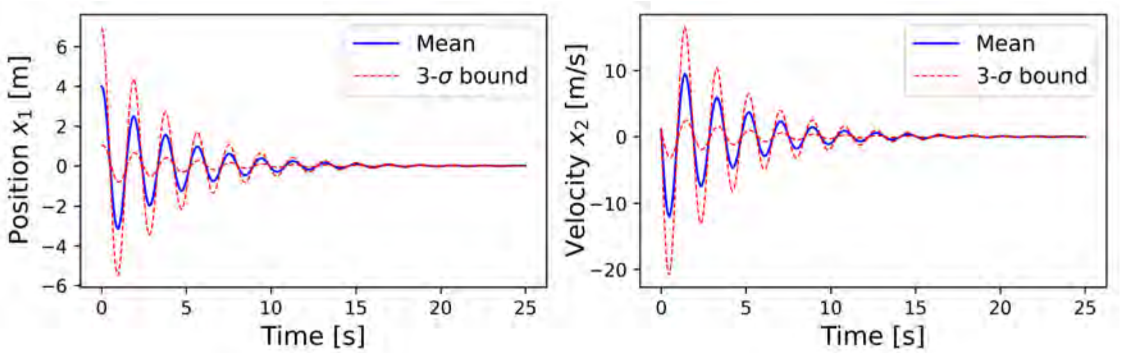


**Figure A3.—**gPC Expansion in mass spring damper system (Case I)

**Table A3.—**Comparison of computational cost (Case I)

| Solution Technique | Number of samples | Computational Time |
|---|---|---|
| gPC solution | 37 | 0.353 $seconds$ |
| Monte Carlo solution | 1000 | 9.59 $seconds$ |
| Computational Resource: 2.3 GHz Intel core i7-11800H CPU | | |

From simulation results, we can see that the 3-$\sigma$ bound for position and velocity solution are large at *t = 0*. It is obvious that a larger number of random variables produce larger uncertainties. As the uncertainty propagates throughout the system, the 3-$\sigma$ bound decreases. Compared to the Monte Carlo technique (computational cost = 9.59 seconds), gPC is computationally efficient (computation cost = 0.353 seconds) and converges to the solution faster.

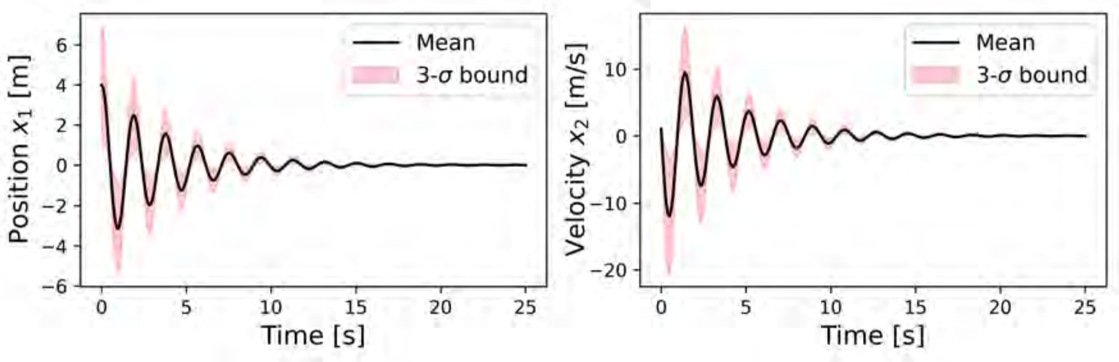


**Figure A4.—** Monte Carlo simulation in mass spring damper system (Number of samples = 1000) Case I

### Case II - Uncertain Initial Conditions and Parameters

For case II, let us consider parameters $m$, $c$, and $k$ are random variables with uniform distribution. Initial conditions are assumed to have the same probability distribution as in case I. Polynomial chaos order ($P$) = 3 and accuracy level for MSG quadrature ($L$) = 6 are used for simulation. The solution of position and velocity using the Python toolbox is shown in Fig. A5. Similarly, Fig. A6 shows the solution using the Monte Carlo technique.

**Table A4.—**Uncertain parameters and their PDF (Case II)

| Variable | Distribution | PDF |
|---|---|---|
| Mass ($m$) | Uniform | $U$ (3.5,4.5) $Kg$ |
| Damping Coefficient ($c$) | Uniform | $U$ (1.5,2.5) $Ns/m$ |
| Spring Constant ($k$) | Uniform | $U$ (43,47) $N/m$ |

**Table A5.—**Comparison of computational cost (Case II)

| Solution Technique | Number of samples | Computational Time |
|---|---|---|
| gPC solution | 315 | 3.198 $seconds$ |
| Monte Carlo solution | 1000 | 10.15 $seconds$ |
| Computational Resource: 2.3 GHz Intel core i7-11800H CPU | | |

We can see that the 3-$\sigma$ bound of position and velocity solution at $t$ = 0 for the stochastic system described in case II is larger compared to the case I. In addition, the 3-$\sigma$ bound decreases slowly with time compared to the case I. From the above plots, we can infer that the Python toolbox gives the same solution as the Monte Carlo simulation with less computational burden. The comparison of computational cost between the proposed toolbox and the Monte Carlo technique shows that the proposed toolbox is dependable, computationally efficient, and can give converged solutions in less time.

## Application to Uncertainty Propagation for sUAV

As an application, we consider a fixed-wing small unmanned aerial vehicle (sUAV) whose equations of motion are described in Section 3.1

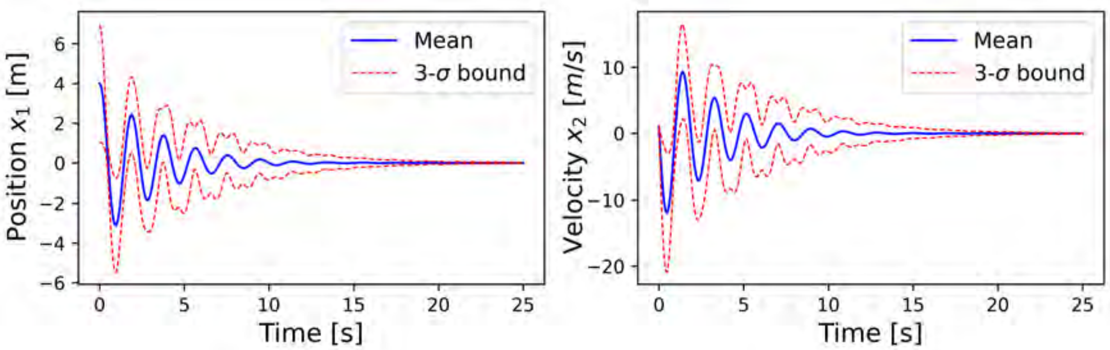


**Figure A5.—** gPC Expansion in mass spring damper system (Case II)

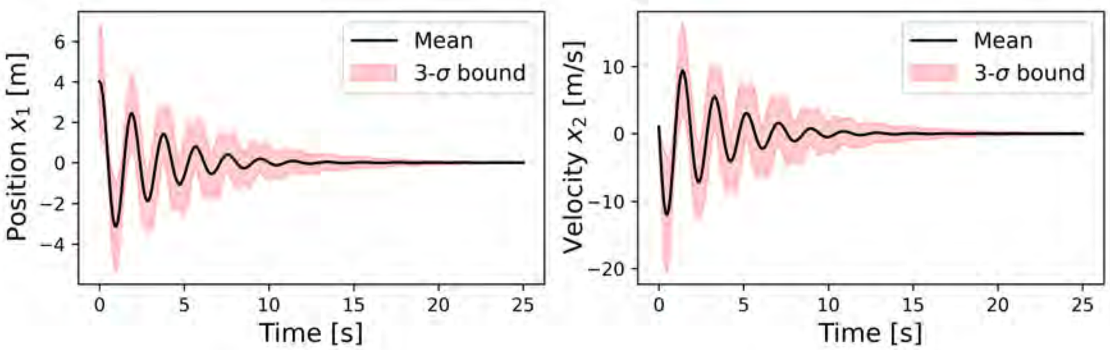


**Figure A6.—**Monte Carlo simulation in mass spring damper system (Number of samples = 1000) (Case II)

**Case I - Smaller uncertainty**

Let us consider an sUAV whose equations of motion are given by Eq. 27 and consider some parameters as random variables. The random parameters and their PDF are given in Table A6. Similarly, other parameters with deterministic values used for simulation are given in Table A7.

Let, the polynomial chaos order ($P$) = 5, accuracy level for MSG quadrature ($L$) = 7 and number of collocation nodes = 747. We propagated uncertainty throughout the model using the proposed toolbox. The solution developed by the toolbox is shown in Fig. A7.

**Table A6.—**Uncertain variables and their PDF (Case I)

| Variables | Distribution | PDF |
|---|---|---|
| $x(0)$ | Normal | $N$(0,0.01$^2$) $m$ |
| $y(0)$ | Normal | $N$(0,0.01$^2$) $m$ |
| $m$ | Uniform | $U$(3,3.25) $Kg$ |
| $C_{D0}$ | Uniform | $U$(0.04,0.055) |
| $k$ | Uniform | $U$(0.0175,0.0225) |
| $S$ | Uniform | $U$(0.41,0.615) $m^2$ |
| $T_{max}$ | Uniform | $U$(2.75,3.5) $N$ |

**Table A7.—**Deterministic parameters

| Parameters | Values | Parameters | Values |
|---|---|---|---|
| $H(0)$ | 250 m | $\eta_c$ | 0.4 |
| $E(0)$ | 270 m | $a_y(0)$ | 0.1$g$ |
| $g$ | 9.81 $m/s^2$ | $a_{yc}$ | 0.1$g$ |
| $\rho$ | 1.225 $Kg/m^3$ | $\eta(0)$ | 1 |
| $\gamma_0$ | 0 rad | $\lambda_n$ | 0.15 |
| $\chi(0)$ | $\pi$/3 rad | $\lambda_p$ | 0.15 |
| $a_p(0)$ | $g$ | $\lambda_y$ | 0.15 |

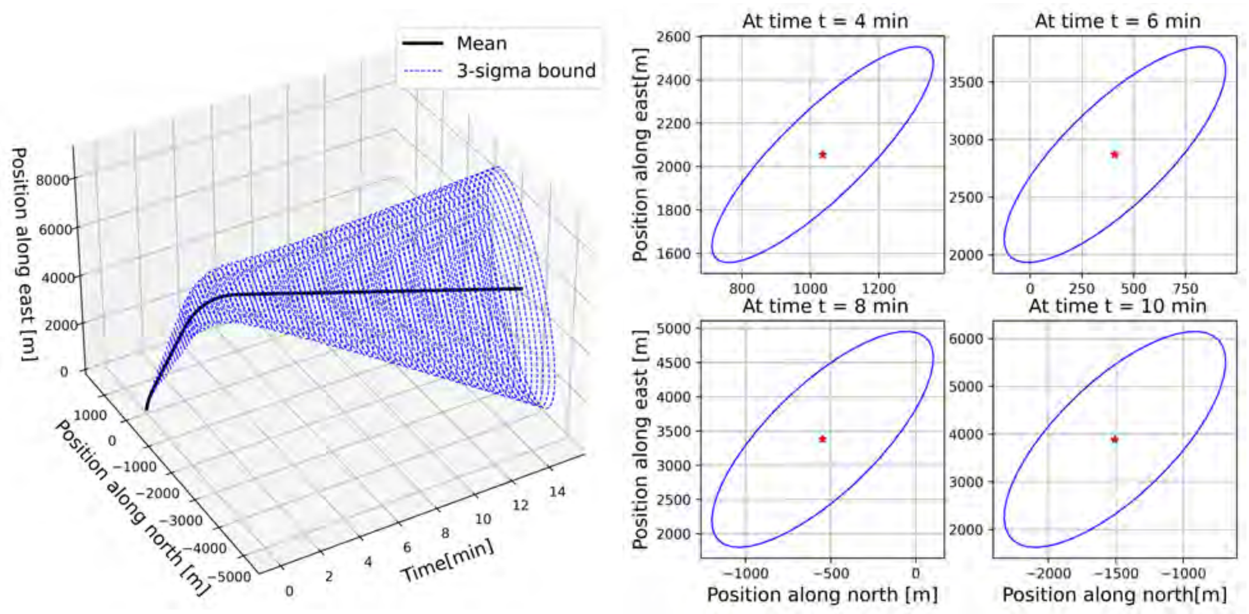


**Figure A7.—**Uncertainty Propagation in sUAV

In Fig. A7, we can observe that the $3 - \sigma$ bound in the trajectory of sUAV increases with time. The covariance error ellipse at four different times instant shows increasing ellipse size along with an increase in simulation time.

### Case II - Larger uncertainty

For case II, let us consider larger uncertainties in three random parameters, as given in Table A8. Polynomial chaos order ($P$) and accuracy level for MSG quadrature ($L$) have the same values as in case I.

**Table A8.—**Uncertain variables and their pdf (Case II)

| Variable | Distribution | PDF |
|---|---|---|
| $m$ | Uniform | $U$(3,4.25) $Kg$ |
| $C_{D0}$ | Uniform | $U$(0.04,0.065) |
| $k$ | Uniform | $U$(0.0175,0.0225) |
| $S$ | Uniform | $U$(0.41,0.615) $m^2$ |
| $T_{max}$ | Uniform | $U$(2.75,5) $N$ |

From simulation results shown in Fig. A8, we can see that the $3 - \sigma$ bound in the trajectory of sUAV has increased. Also, the error ellipse at different time instants is larger than in case I. Thus, we can infer that the increase in parameter uncertainty enlarges the $3 - \sigma$ bound.

### Case III - Uncertainty propagation in multiple sUAVs

Let us consider four sUAV flying in airspace. Let the parametric uncertainty and values for polynomial chaos order ($P$) and accuracy level for MSG quadrature ($L$) be the same as in case I. The PDF of uncertain

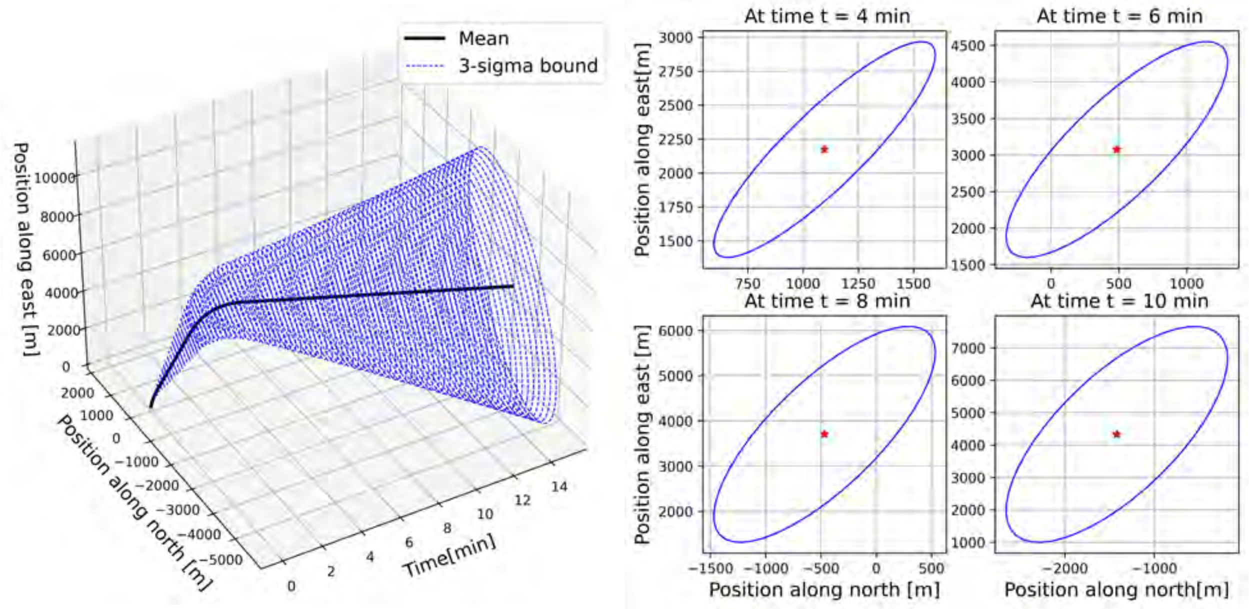


**Figure A8.—**Uncertainty Propagation in sUAV (Case II)

initial conditions is in Table A9. The initial heading angles of each sUAVs are $\chi_1 = 60°$, $\chi_2 = 315°$, $\chi_3 = 225°$, and $\chi_4 = 90°$.

**Table A9.—**Uncertain variables and their PDF (Case III)

| **Initial Conditions** | **Distribution** | **PDF** |
|---|---|---|
| $x_1(0)$ | Normal | $N$(0,0.01$^2$) $m$ |
| $y_1(0)$ | Normal | $N$(0,0.01$^2$) $m$ |
| $x_2(0)$ | Normal | $N$(0,0.01$^2$) $m$ |
| $y_2(0)$ | Normal | $N$(8000,100$^2$) $m$ |
| $x_3(0)$ | Normal | $N$(15000,100$^2$) $m$ |
| $y_3(0)$ | Normal | $N$(6500,100$^2$) $m$ |
| $x_4(0)$ | Normal | $N$(14000,100$^2$) $m$ |
| $y_4(0)$ | Normal | $N$(0,0.01$^2$) $m$ |

In Fig. A9, the open loop simulation of the sUAVs flight path shows that the sUAVs have a risk of conflict after 10 minutes of flight. Thus, to prevent a conflict, an adjustment in flight path is necessary, i.e., a change in separation between each sUAVs. Table A11 shows the computational cost for uncertainty propagation of all sUAVs. A comparison of computational cost between the Monte Carlo solution ( computational cost = 978.672 seconds) and the gPC solution (computational cost = 59.8325 seconds) shows that gPC converges to the solution with less computational burden.

**Table A10.—** Deterministic parameters (Case III)

| **For sUAV-1 and sUAV-3** |
|---|
| $$a_{yc} = \begin{cases} 0, & \text{t} \leq 4mins \\ -0.01g, & 4mins < \text{t} < 5.5mins \\ 0, & else \end{cases}$$ |
| **For sUAV-2 and sUAV-4** |
| $$a_{yc} = \begin{cases} 0, & \text{t} \leq 4mins \\ 0.01g, & 4mins < \text{t} < 5.5mins \\ 0, & else \end{cases}$$ |

**Table A11.—**Computational cost for uncertainty propagation in sUAV.

| Case | Solution Technique | Number of samples | Computational Time |
|---|---|---|---|
| I | gPC Solution | 747 | 14.7094 *seconds* |
| | Monte Carlo Solution | 10000 | 272.9157 *seconds* |
| II | gPC Solution | 747 | 14.8759 *seconds* |
| | Monte Carlo Solution | 10000 | 275.62 *seconds* |
| III | gPC Solution | 747 | 59.8325 *seconds* |
| | Monte Carlo Solution | 10000 | 978.672 *seconds* |
| Computational Resource: 2.3 GHz Intel core i7-11800H CPU | | | |

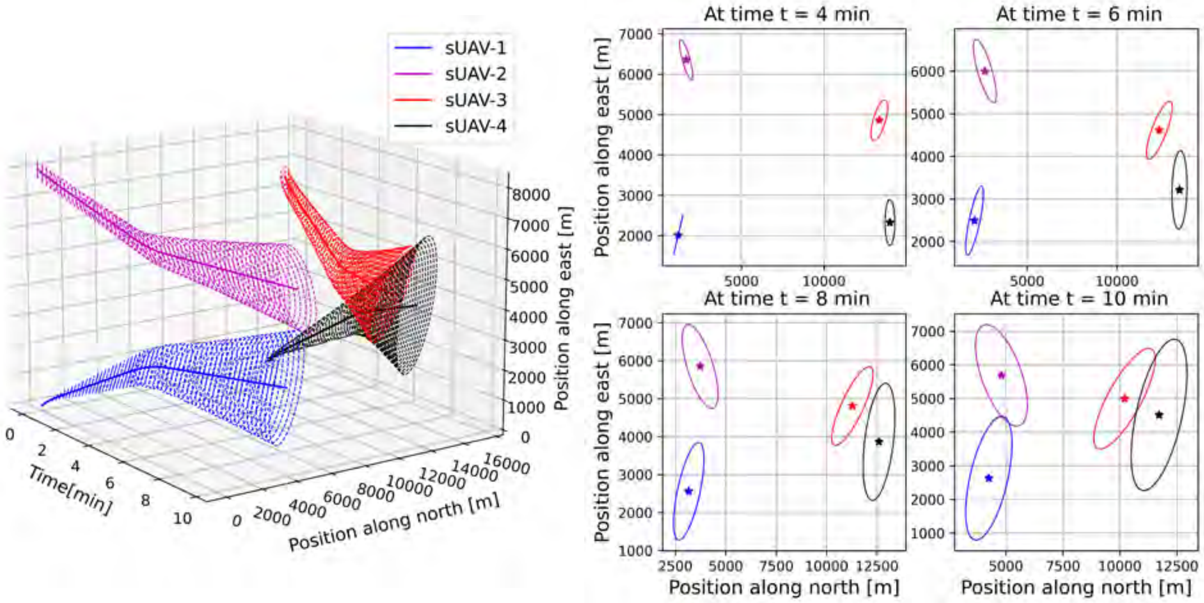


**Figure A9.—**Uncertainty Propagation in multiple sUAVs (Case-III)

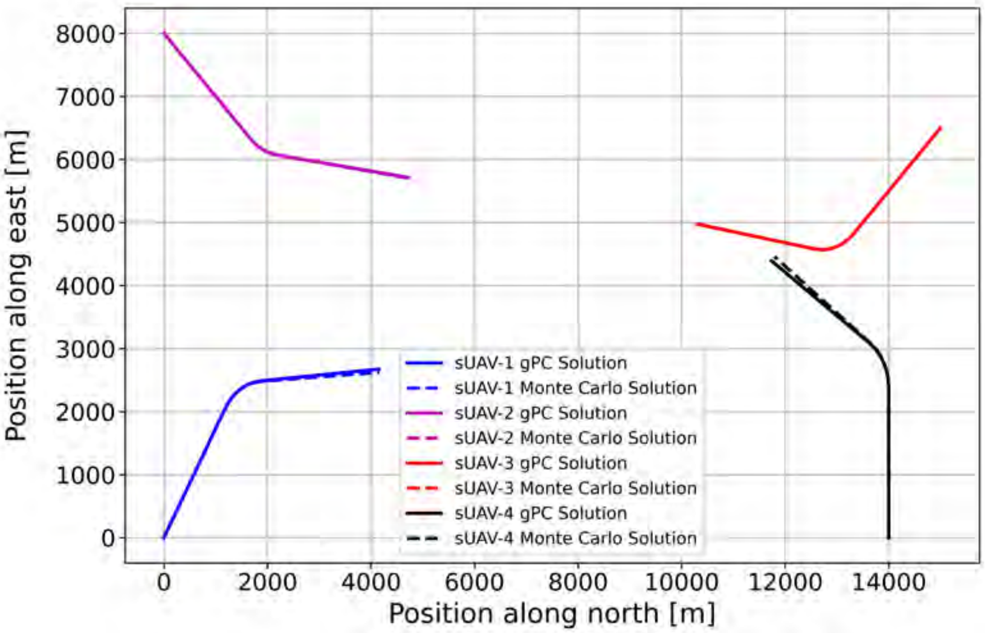
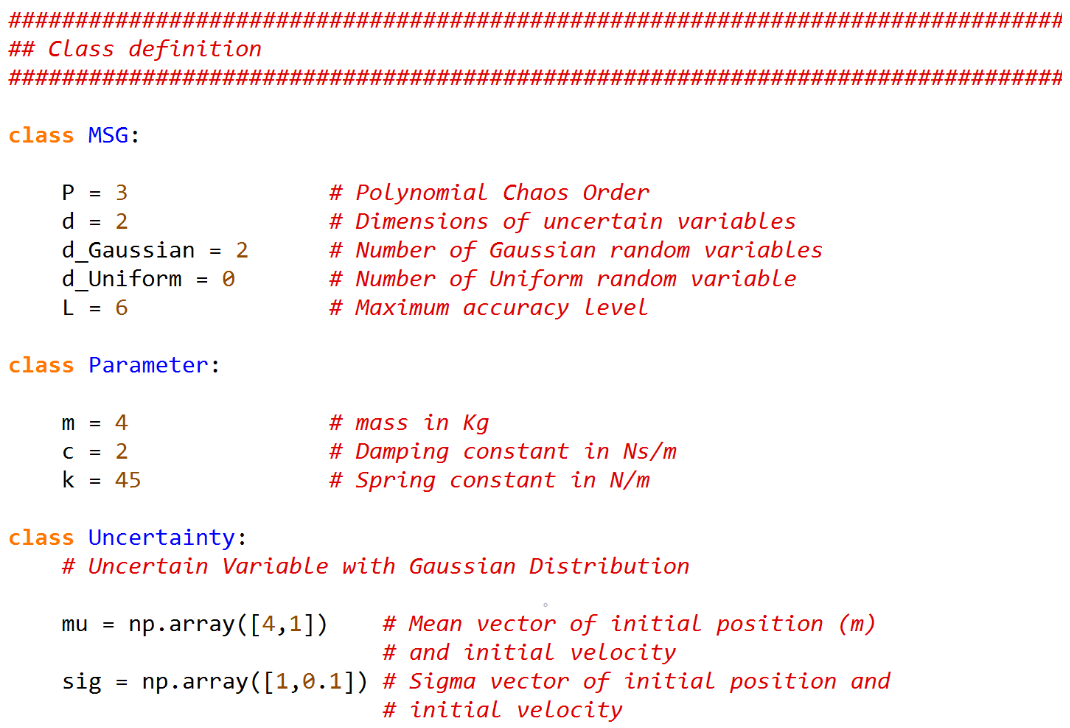


**Figure A10.—** gPC and Monte Carlo Solution for mean trajectory (Case-III)

### Illustrative Code

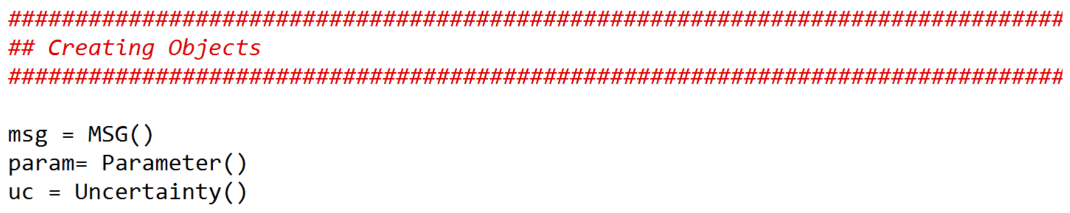
In this subsection, we have given an illustrative code for *Main module* of mass spring damper system with initial conditions uncertain. Readers can refer to this section for creating a *Main module* for the problem of their interest.

Pre-processor section



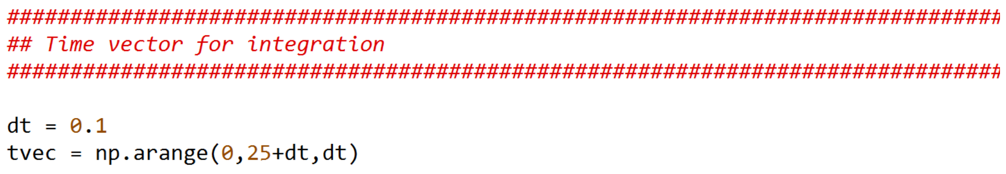
```
###############################################################################
## Class definition
###############################################################################

class MSG:

    P = 3                   # Polynomial Chaos Order
    d = 2                   # Dimensions of uncertain variables
    d_Gaussian = 2          # Number of Gaussian random variables
    d_Uniform = 0           # Number of Uniform random variable
    L = 6                   # Maximum accuracy level

class Parameter:

    m = 4                   # mass in Kg
    c = 2                   # Damping constant in Ns/m
    k = 45                  # Spring constant in N/m

class Uncertainty:
    # Uncertain Variable with Gaussian Distribution

    mu = np.array([4,1])    # Mean vector of initial position (m)
                            # and initial velocity
    sig = np.array([1,0.1]) # Sigma vector of initial position and
                            # initial velocity
```

Define class $MSG$, $Parameter$ and $Uncertainty$.



```
###############################################################################
## Creating Objects
###############################################################################

msg = MSG()
param= Parameter()
uc = Uncertainty()
```

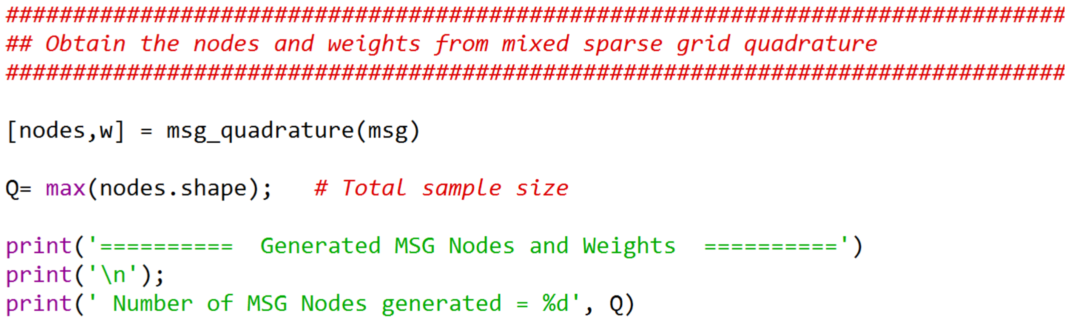
Create objects *msg, param, and uc* of the defined class.



```
###############################################################################
## Time vector for integration
###############################################################################

dt = 0.1
tvec = np.arange(0,25+dt,dt)
```

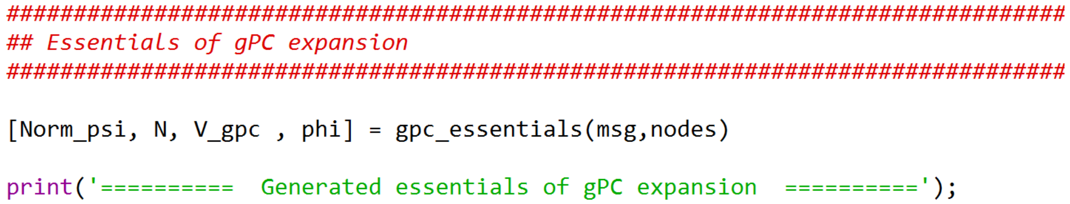
Define $tvec$ → time vector with step size of 0.1 → an array of size (251,1)

Processor/Solver section



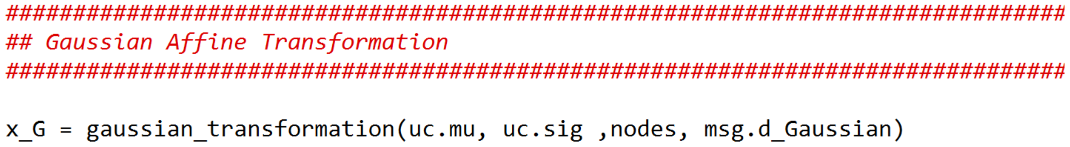
```
##################################################################################
## Obtain the nodes and weights from mixed sparse grid quadrature
##################################################################################

[nodes,w] = msg_quadrature(msg)

Q= max(nodes.shape);   # Total sample size

print('==========  Generated MSG Nodes and Weights  ==========')
print('\n');
print(' Number of MSG Nodes generated = %d', Q)
```

Call function *msg_quadrature* and pass *msg* object as the argument. This function returns nodes and weights using a mixed sparse grid technique.
$nodes$ → Collocation Nodes → an array of size (37,2)
$w$ → Node weights → an array of size (37,1)



```
##################################################################################
## Essentials of gPC expansion
##################################################################################

[Norm_psi, N, V_gpc , phi] = gpc_essentials(msg,nodes)

print('==========  Generated essentials of gPC expansion  ==========');
```

Call *gpc_essentials function* and pass *msg* object and nodes as the argument. This function returns the normalization coefficient, gPC order, and indexes.
$Norm\_psi$ → Normalization Coefficient → an array of size (1,10)
$N$ → gPC order → 10
$V\_gpc$ → combine $syschar$ (string representation of the number of the orthogonal polynomial for underlying random variables) and $indexes$ (vector of indexes → an array of size (10,2)) as one variable → list of size 2



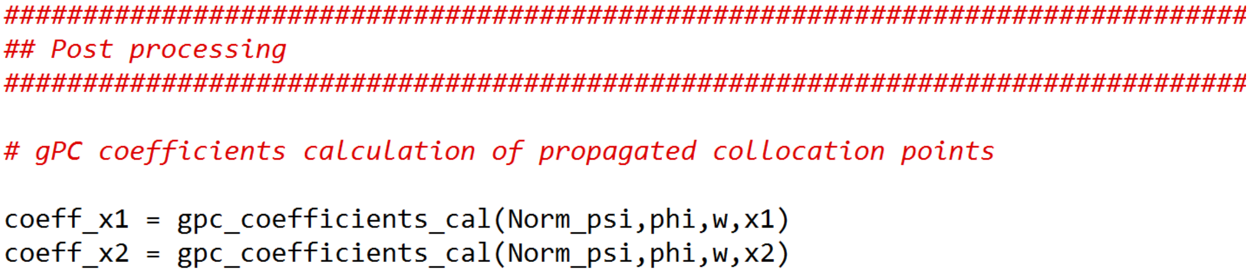
```
##################################################################################
## Gaussian Affine Transformation
##################################################################################

x_G = gaussian_transformation(uc.mu, uc.sig ,nodes, msg.d_Gaussian)
```

$x\_G$ → stores initial conditions transformed to normally distributed collocation nodes with a mean and standard deviation of uncertain variables → array of size (37,2)

```
################################################################################
## Obtain the solution of quantities of interest
################################################################################

# Buffer
x1 = np.zeros([len(tvec),Q])
x2 = np.zeros([len(tvec),Q])

for cc in range(0,Q):

    X0 = [x_G[cc,0],x_G[cc,1]]     # Initial Conditions
    X1 = solve_ivp(ode,[tvec[0],tvec[-1]],X0,method='RK45',t_eval = tvec,\
                   args=(param,),vectorized = True)

    x1[:,cc] = X1.y[0,:]  # First state
    x2[:,cc] = X1.y[1,:]  # Second state
```

Solving differential equations at collocation nodes using Runge-Kutta 45 solver.
$X\_0$ → vector of initial condition at single collocation node → list of size 2
$x1$ → stores position at each collocation node throughout the time vector → array of size (251,37)
$x2$ → stores velocity at each collocation node throughout the time vector → array of size (251,37)

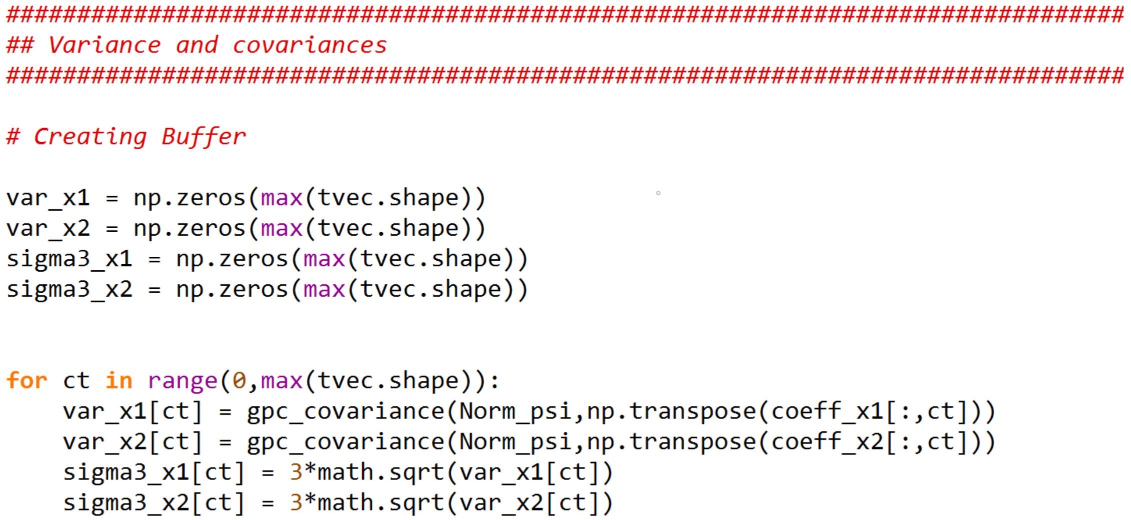
Post processor section



```
################################################################################
## Post processing
################################################################################

# gPC coefficients calculation of propagated collocation points

coeff_x1 = gpc_coefficients_cal(Norm_psi,phi,w,x1)
coeff_x2 = gpc_coefficients_cal(Norm_psi,phi,w,x2)
```

Call *gpc_coefficients_cal function* and pass variables $Norm\_psi$, $phi$, $w$ , $x1$, and $x2$ as arguments.
$coeff\_x1$ → coefficient of gPC expansion for position solution → an array of size (10,251)
$coeff\_x2$ → coefficient of gPC expansion for velocity solution → an array of size (10,251)

```
################################################################################
# Mean and Variances Calculation
################################################################################
# Mean
################################################################################

x1_mean = coeff_x1[0, :]
x2_mean = coeff_x2[0, :]
```



```
#################################################################################
## Variance and covariances
#################################################################################

# Creating Buffer

var_x1 = np.zeros(max(tvec.shape))
var_x2 = np.zeros(max(tvec.shape))
sigma3_x1 = np.zeros(max(tvec.shape))
sigma3_x2 = np.zeros(max(tvec.shape))


for ct in range(0,max(tvec.shape)):
    var_x1[ct] = gpc_covariance(Norm_psi,np.transpose(coeff_x1[:,ct]))
    var_x2[ct] = gpc_covariance(Norm_psi,np.transpose(coeff_x2[:,ct]))
    sigma3_x1[ct] = 3*math.sqrt(var_x1[ct])
    sigma3_x2[ct] = 3*math.sqrt(var_x2[ct])
```

$x1\_mean$ → position mean → first row of gPC coefficient → an array of size (251,1)
$x2\_mean$ → velocity mean → first row of gPC coefficient → an array of size (251,1)

Calculating covariance
Call *gpc_covariance function* and pass variables $Norm\_psi$, $coeff\_x1$, and $coeff\_x2$ as arguments.
$var\_x1$ → position variance → an array of size (251,1)
$var\_x2$ → velocity variance → an array of size (251,1)

## Toolbox Repository

The Python toolbox can be downloaded from `https://github.com/aslResearch/aslUQ/tree/master/gPC-Python-Toolbox`