2019 International Conference on Unmanned Aircraft Systems (ICUAS)
Atlanta, GA, USA, June 11-14, 2019

Modeling Unmanned Aerial System (UAS) Risks via Monte Carlo Simulation

Eliot Rudnick-Cohen[1], Jeffrey W. Herrmann[1,2], Shapour Azarm[1]

***Abstract*— Unmanned Aerial Systems (UAS) pose a variety of risks to third parties when operating over populated areas, due to the danger posed if the UAS crashes. Two commonly used metrics for assessing the risk of such crashes are the kinetic energy of the UAS at the time of impact and the probability distribution of locations where the UAS could crash. In this paper, a Monte Carlo based approach is presented for simulating UAS crashes in order to calculate these metrics. A surrogate modeling approach for UAS safety metrics is also presented, which is built using the results of the Monte Carlo simulations. The surrogate modeling approach is capable of rapidly evaluating UAS safety metrics for arbitrary UAS design and operating parameters. The proposed approach is demonstrated by modeling the kinetic energies at time of impact and crash probability distributions for UAS with dynamics models similar to that of a Cessna 182.**

I. INTRODUCTION

As Unmanned Aerial Systems (UASs) are increasingly adopted across numerous domains, assessing the safety of operating UASs over people is becoming crucial for increasing public acceptance of such operations. This includes assessing and mitigating the risk posed by the UAS if it were to crash in a populated area. Performing this assessment requires understanding where the UAS is likely to crash after a failure and its kinetic energy when it crashes into the ground. Determining these quantities empirically is impractical, as it would require repeatedly crashing a UAS. Predicting these quantities is also challenging because the dynamics model of a UAS is a non-linear model (like that of any aircraft) and is sensitive to its initial conditions. This makes it difficult to estimate the risks posed by a UAS crash analytically without simplifying the dynamics model or ignoring the dynamics in an overly conservative risk assessment.

This paper presents a simulation-based approach that can generate results and surrogate models to support assessing the safety of UAS operations over populated areas in practical settings. The approach is based on a Monte Carlo simulation of a flight dynamics model that simulates the trajectories of a UAS crashing from different initial flight states. These trajectories produce a probability distribution of the crash location, where the UAS reaches the ground, and can also be used to estimate the UAS's kinetic energy when it impacts the ground. The approach performs simulation experiments to generate the distributions for different values of UAS design and operating parameters. These distributions provide insights into how these parameters affect the Crash location Probability Distribution (CPD) and kinetic energy. Finally, the approach generates surrogate models from these results that can be used to quickly estimate the CPD and kinetic energy for a given set of parameter values. This paper also discusses the results and surrogate models generated by using the proposed approach to model UAS with dynamics similar to a Cessna 182 [1]

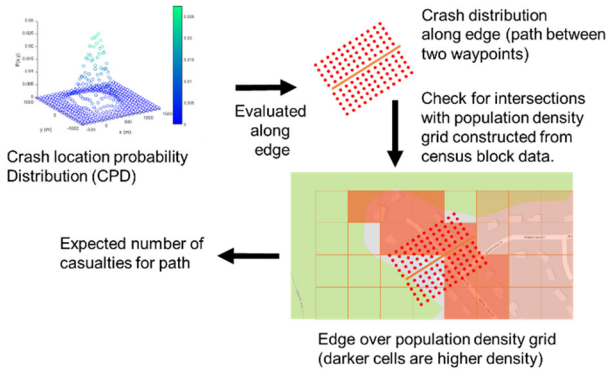

Fig. 1. Example of how a Crash Probability Distribution (CPD) is used to assess the expected number of fatalities of a UAS operation

The remainder of this paper is organized as follows. Section II reviews previous work on assessing UAS safety and positions this paper with respect to related work. Section III presents the proposed Monte Carlo simulation-based approach and the approach for generating surrogate models. Section IV applies the proposed Monte Carlo simulation approach to perform a parametric study involving design and operating parameters for UAS with similar dynamics to a Cessna 182. Section V presents a statistical analysis of the results from Section IV to determine which parameters cause statistically significant effects on the safety of the example UAS model. Section VI tests the performance of the surrogate models built using the proposed approach. Section VII summarizes the results presented.

II. RELATED WORK

Multiple methods [2] and frameworks [3] have been proposed for assessing UAS safety. One commonly used quantity in UAS risk assessment is the expected number of fatalities associated with a UAS operation, which requires

[1]Department of Mechanical Engineering, University of Maryland, College Park, MD 20742 {erudnick, azarm, jwh2}@umd.edu
[2]Institute for Systems Research, University of Maryland, College Park, MD 20742

assessing the probability of failure during an operation and estimating the number of persons struck by a UAS if it were to crash. This estimate requires a CPD (see Figure 1) to determine how many people could be affected by a UAS crash. The expected fatalities caused by a UAS flight is especially relevant for analyzing the safety of a large UAS, which is likely to cause a fatality if it strikes someone due its high mass and velocity [4]. Approaches for planning a UAS's path to mitigate this risk [5, 6, 7] assess the expected number of fatalities by moving the CPD along a flight path, as depicted in Figure 1.

There are several ways in which CPDs are constructed. Some approaches [7, 8, 9, 10, 11, 12] use analytical models to determine the area of all possible locations which a UAS could reach before crashing and then assume that it is equally likely that UAS crashes in any of these locations. La Cour Harbo [13], La Cour Harbo and Schioler [14], and Primatesta et al. [7] used the distance traveled to develop a non-uniform CPD model and also consider the effects of wind. Lum et al. [15] used Monte Carlo simulation to determine the distance traveled by a UAS before crashing, which is used to fit an analytical CPD model. Rudnick-Cohen et al. [5] used Monte Carlo simulation of a UAS crashing to construct the CPD out of the binned frequency distribution of the locations of the simulated crashes.

Prior work has explored how UAS CPDs are affected by either different operating parameters or different design configurations. Wu and Clothier [9] analytically determined the shapes of the regions of all possible crash locations for failures occurring at different altitudes. Rudnick-Cohen et al. [16] constructed a Delaunay triangulation from the results of Monte Carlo simulations to model the effects of design parameters on a CPD to perform design optimization. Haartsen et al. [17] conducted a parameter study on how the operating parameters such as flight speed, altitude, and the vehicle's roll angle affect the CPD of fixed wing and rotorcraft UAS.

Another quantity used to assess UAS safety is the impact kinetic energy of the UAS when it crashes into the ground, which determines whether a UAS crash can cause fatalities. The U.S. Federal Aviation Administration (FAA) has proposed preliminary requirements for small UAS operating over people that require the kinetic energy at time of UAS impact be below either 11 or 25 ft-lbs, depending on where the UAS is being flown [18]. These limits on kinetic energy originated from a U.S. Army Range Commander Council specification [4, 19]. The kinetic energy depends upon the mass and the velocity of the UAS. Estimating the UAS's velocity as it crashes requires a model of the UAS's flight dynamics, which are affected by the UAS's physical design and its flight state at time of failure. This makes it difficult to directly compute the kinetic energy of a UAS at the time it crashes into the ground.

The approach presented in this paper expands upon the Monte Carlo approach introduced in [5] with the following extensions: (1) an integrated simulation-based approach for estimating the CPD and kinetic energy of a crashing UAS; (2) insights into how changes to design and operating parameters affect the CPD and kinetic energy; and (3) surrogate models that can be used to quickly estimate these safety metrics for UAS.

TABLE I: Variables used in Table II

U, V, W	Velocities (Body Frame) (m/s)
θ, ϕ, ψ	Orientation (Euler angles) (rad)
P, Q, R	Angular velocities (Body Frame) (rad/s)
p_N, p_E, h	Position of vehicle (NED Frame) (m)
F_x, F_y, F_z	Drag forces on vehicle (Body Frame) (N)
$\bar{L}, M, N$	Drag moments on vehicle (Body Frame) (N)
$c_1, c_2, ..., c_9$	Moment of inertia constants (kg/m^2)
g	Gravitational constant (m/s^2)

III. MONTE CARLO SIMULATION OF UAS CRASHES AND SURROGATE MODELING OF UAS SAFETY METRICS

A. Simulation Model

Table I defines the variables in our model of the dynamics of an air vehicle in steady state flight, which can be described using the ordinary differential equations (ODEs) in Table II [1]. By solving these ODEs for different initial conditions, different crash trajectories will be obtained. By changing these initial conditions and performing Monte Carlo simulation, a range of potential crash trajectories will be generated. Note that the forces and moments present in Table I will have different values and equations depending on the type of UAS being modeled. This can require modeling additional state variables to those in Table I, such as control surfaces (fixed wing UAS) or rotors (multicopter UAS).

TABLE II: Flight dynamics equations for Monte Carlo crash simulations, overdots denote time derivatives

$$\dot{U} = RV - QW - g\sin(\theta) + \frac{F_x}{m}$$

$$\dot{V} = -RU + PW + g\sin(\phi)\cos(\theta) + \frac{F_y}{m}$$

$$\dot{W} = QU - PV + g\cos(\phi)\cos(\theta) + \frac{F_z}{m}$$

$$\dot{\phi} = P + \tan(\theta)(Q\sin(\theta) + R\cos(\phi))$$

$$\dot{\theta} = Q\cos(\phi) - R\sin(\phi)$$

$$\dot{\psi} = \frac{Q\cos(\phi) + R\sin(\phi)}{\cos(\theta)}$$

$$\dot{P} = (c_1 R + c_2 P)Q + c_3 \bar{L} + c_4 N$$

$$\dot{Q} = c_5 PR - c_6 (P^2 - R^2) + c_7 M$$

$$\dot{R} = (c_8 P - c_2 R)Q + c_4 \bar{L} + c_9 N$$

$$\dot{p}_N = U\cos(\theta)\cos(\psi) + V(-\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi)) + W(\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi))$$

$$\dot{p}_E = U\cos(\theta)\sin(\psi) + V(\cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\cos(\psi)) + W(-\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi))$$

$$\dot{h} = U\sin(\theta) - V\sin(\phi)\cos(\theta) - W\cos(\phi)\cos(\theta)$$

In order to keep the Monte Carlo simulations computationally tractable, we ignored any effects on drag coefficients caused by angular velocities (P, Q, R) and the time derivatives of the UAS's angle of attack and sideslip angle. We found that this does not significantly affect the location where the UAS crashes, but it provides a large reduction in the time needed to simulate the UAS crashing.

B. *Monte Carlo Approach*

Let $x(t)$ be the state of the UAS as a function of time t, and let $\vec{x}^r$ be the initial state. Let $f(x)$ be the equation of the flight dynamics such that $\dot{x} = f(x(t))$. Let X^d be the range of the random perturbations. Let t_{fi} be the initial amount of integration time the ODE is solved for. Let $\mathrm{ME}(x)$ be a function that returns the total mechanical energy (kinetic energy plus potential energy) of state x. Let N be the number of iterations. Table III details the algorithm used to perform the Monte Carlo simulation. Although the flight dynamics model $f(x)$ is deterministic, each iteration of the simulation uses randomly perturbed initial conditions (see Table III, steps 1.a and 1.b). The equations in Table I can be solved for specific initial conditions using an ODE solver (see Table III, step 1.d.ii). However, most efficient ODE solvers (e.g., MATLAB's ode45 [20]) make use of adaptive step sizes, which complicates identifying the exact time and location at which a vehicle crashes. Additionally, the amount of time which the UAS remains in the air before crashing could vary significantly. To account for this, we ran the ODE solver for a fixed amount of integration time before checking for a crash (see Table III, step 1.d). If the vehicle's altitude is still above the ground, then the final solution from the previous ODE solution is used as the new initial conditions and the ODE solver's integration time is set to be the total amount of integration time used (see Table III, step 1.d.i). Repeating this process allows for using an appropriate amount of integration time as needed during the crash simulation. When the UAS's altitude is below the ground, we extract the first time in the ODE solution where the UAS is under the ground and use the state at that time as the state when the UAS crashed (see Table III, step 1.e).

TABLE III: Monte Carlo Crash simulation approach

1. While $N > 0$
   a. Let $\vec{x}^d$ be a random perturbation drawn from X^d
   b. $\vec{x}^i = \vec{x}^r + \vec{x}^d$
   c. $t_i = 0, t_f = t_{fi}, t_{end} = 0$
   d. While $\vec{x}^i$ corresponds to a state that hasn't crashed
      i. Set $t_f = t_{end} + t_f, t_i = t_{end}$
      ii. Solve the ODE system $\dot{x} = f(x(t))$ on time interval $[t_i,\ t_f]$, with initial conditions $x(t_i) = \vec{x}^i$, let $\vec{x}^f$ be the final state in the solution and let t_{end} be the time the solver stopped at. The solver should stop before reaching t_f if an event occurs
      iii. Set $\vec{x}^i = \vec{x}^f$
   e. Let t_{crash} the first time in the solution for $x(t)$ where the vehicles height is less than zero. $\vec{x}_{crash} = x(t_{crash})$
   f. If $\mathrm{ME}(\vec{x}^f) < \mathrm{ME}(\vec{x}^r + \vec{x}^d)$
      i. Store $\vec{x}_{crash}$ in the list of valid Monte Carlo simulation results
   g. Set $N = N - 1$
2. Return the list of valid Monte Carlo simulation results and finish

The ODE solver also needs to be stopped if an event that changes the system's dynamics occurs. For example, if a UAS with unpowered (free to move while crashing) control surfaces is simulated, the ODE solver needs to be stopped any time the control surfaces hit a hard stop, as the dynamics of the UAS will have changed. In this case the ODE solver is restarted with the state at the time which the event occurs and the integration time for the next iteration is set as it normally would be. This ensures that the crash simulation avoids any instability that could be caused by the UAS's dynamics changing. From a practical standpoint, an event should be generated once the UAS's altitude reaches a fixed height beneath the ground. This allows the ODE solver to be stopped early, reducing computational time. However, the event should not be generated when the UAS's altitude is zero, as this would interfere with ODE solvers that move forwards and backwards in time (e.g., MATLAB's ode45 [20]).

Because the Monte Carlo simulation randomly perturbs the initial conditions of the UAS, many different trajectories are generated for a single initial state. The CPD generated from this initial state is the 2-D binned histogram of the locations where the UAS reached the ground. The maximum kinetic energy at time of impact is determined from the greatest impact velocity observed over the N iterations.

However, care needs to be taken to account for degenerate results, which are caused by initial conditions for which the ODE solver is unstable. Fortunately, such cases are easy to identify, as the final kinetic energy of the UAS will often increase to a quantity larger than the initial mechanical energy in the system. Although the initial velocities of the UAS are typically variables perturbed in the Monte Carlo simulation, this check (Table III, step 1.f) can be done against the unperturbed velocity, since drag forces should cause a significant loss in the UAS's mechanical energy.

C. *Using parameter studies to construct surrogate models*

In order to investigate the effects of various UAS design or operation parameters, the Monte Carlo approach from Section III.B can be run for multiple parameter configurations. A large-scale parameter study with an appropriate design of experiments (DOE), such as a Full Factorial or Latin Hypercube design, will generate enough

CPDs such that CPDs for other parameter configurations can be interpolated from them.

A k-Nearest Neighbors (KNN) model can be used to interpolate the results of multiple Monte Carlo simulations to obtain the kinetic energy at time of crashing and the CPD of a UAS for a queried set of UAS parameters. The KNN model performs an inverse distance weighted average of the Monte Carlo simulation results for the k design points (parameter configurations in the DOE) nearest to the queried parameters, where the distance $d(x, y)$ is determined by the weighted distance measure given in Eq. (1). In this distance measure, w denotes the weight for each parameter, x denotes the input queried parameters, and y denotes the parameters of one of the design points.

$$\begin{aligned} d(x,y) = {} & w_{height}(x_{height} - y_{height})^2 + w_{roll}(x_{roll} - y_{roll})^2 + \\ & w_{pitch}(x_{pitch} - y_{pitch})^2 + w_{speed}(x_{speed} - y_{speed})^2 + \\ & w_{wingspan}(x_{wingspan} - y_{wingspan})^2 + \\ & w_{rudder}(x_{rudder} - y_{rudder})^2 + w_{mass}(x_{mass} - y_{mass})^2 \end{aligned} \quad (1)$$

Eq. (1) is used with an inverse distance weighting [21] in order to combine together the results from the k-nearest neighbors to the queried parameters. The inverse distance weighting exponent u, the weights w and the number of neighbors k should be separately determined for each UAS safety metric being modeled. The performance of the KNN model is tuned by adjusting the weights w, the number of neighbors k and the inverse distance exponent u.

IV. EXAMPLE: UAS BASED OFF CESSNA 182

To demonstrate the Monte Carlo crash simulation approach, we tested it on a UAS with the dynamics of a Cessna 182 suffering from a total loss of power, meaning that no systems on the UAS would be actuated during the crash. The Cessna 182's dynamics model was chosen because its aerodynamic coefficients are publicly available [1]. A 250 element Latin Hypercube design of experiments (DOE) was created using MATLAB [22] and Monte Carlo simulations were run for each of the 250 experiments in the DOE. 7 parameters were used for the DOE, the UAS's height or elevation from the ground level, roll, pitch and forwards velocity at time of failure and its wingspan, allowable rudder movement range and mass. The ranges of these parameters are listed in Table IV.

A. *Cessna 182 based UAS dynamics model including unactuated control surfaces*

The UAS dynamics considered were those of a Cessna 182 with unactuated control surfaces, which were modeled as simple masses with a mass of 1 kg subject to the drag forces caused by the control surfaces. The Cessna 182 model has three types of control surfaces, ailerons, elevators, and a rudder. We assume that the elevator control surfaces move together and that the aileron control surfaces always move in opposite directions to each other. Each control surface has a maximum movement range; a constraint in the ODE dynamics model prevents it moving outside this range. If a control surface reaches this limit, the ODE solver is stopped and restarted to account for the change in the dynamics.

TABLE IV: Parameter ranges used for Latin Hypercube DOE

	Height (m)	Roll	Pitch	Speed (m/s)	Wingspan (m)	Rudder Range	Mass (kg)
Min	512	-45°	-15°	30	7.45	±24°	887
Max	2024	45°	15°	70	19.47	±40°	1400

TABLE V: INITIAL CONDITIONS FOR MONTE CARLO SIMULATIONS

Velocity (m/s)	Perturbation
U	10
V	10
W	10
Position (m)	
p_N	0
p_E	0
h	0
Orientation, Euler angles (degrees)	
Φ (Roll)	11.25
Θ (Pitch)	11.25
Ψ (Yaw)	11.25
Angular Velocity (degrees/s)	
P	11.25
Q	11.25
R	11.25
Control surface deflection (degrees)	
Elevator Deflection (δ_E)	11.25
Rudder Deflection (δ_R)	11.25
Aileron Deflection (δ_A)	11.25
Control surface deflection rates (degrees/s)	
Elevator deflection rate ($\dot{\delta}_E$)	0
Rudder deflection rate ($\dot{\delta}_R$)	0
Aileron deflection rate ($\dot{\delta}_A$)	0

B. *Parameter settings for example*

The random perturbations used in the Monte Carlo simulations are detailed in Table V. The values for U (speed), h (height), Φ (roll) and Θ (pitch) were perturbed from the values specified for each Monte Carlo simulation by the DOE. All other parameters detailed in Table V were perturbed from a value of zero. Perturbations were applied in both positive and negative directions.

The Monte Carlo simulations used $t_{fi} = 0.25$ seconds, additionally a minimum time step of 10^{-5} seconds was imposed on the ODE solver used. Any time the ODE solver's timestep went below 10^{-5} seconds, an event was generated to stop the ODE solver, to avoid wasting computational time in degenerate cases. At each design point, we ran $N = 10{,}000$ crash simulations, so a total of 2.5 million crash simulations were performed for the entire DOE considered. The ODE solver used was MATLAB's ode45 [20] with absolute and relative tolerances set at 10^{-3}. An event to stop the ODE solver was generated whenever the UAS fell more

than 100 meters beneath the ground (altitude of 0 meter). Additionally, to avoid wasting computational time in cases where the ODE solver became unstable, a time limit of 30 seconds was imposed for each iteration (Step 1, Table III) of the Monte Carlo simulations.

C. *Results from example*

Based on the kinetic energy at the time of crashing, the crash trajectories produced by the Monte Carlo simulations can be grouped into two different types. Figure 2 shows the history of the UAS's velocities for examples of these two types of trajectories. Note that the high frequency of oscillations in Figure 2 occurs because the model ignores rate terms for angular rate; moreover, Figure 2 depicts body frame velocity. The speed of the vehicle does not oscillate at the magnitudes or frequencies of its components depicted in Figure 2. Trajectory (a) corresponds to the UAS crashing by flying along a stable trajectory to the ground, as evidenced by the lack of oscillations in *U*, the body frame velocity component moving directly against the UAS's drag. Trajectory (b) corresponds to the UAS crashing by falling, as indicated by lack of oscillations in W, the velocity component moving directly against the UAS's lift. Because fixed wing aircraft (such as the Cessna 182) are designed to have high L/D (lift over drag) ratios, Trajectory (a) incurs significantly less loss in speed that Trajectory (b), causing Trajectory (a) to have a much higher kinetic energy when it crashes into the ground. This implies that design changes that increase drag (such as increasing wingspan) should also provide reductions in the kinetic energy of a fixed wing UAS when it crashes.

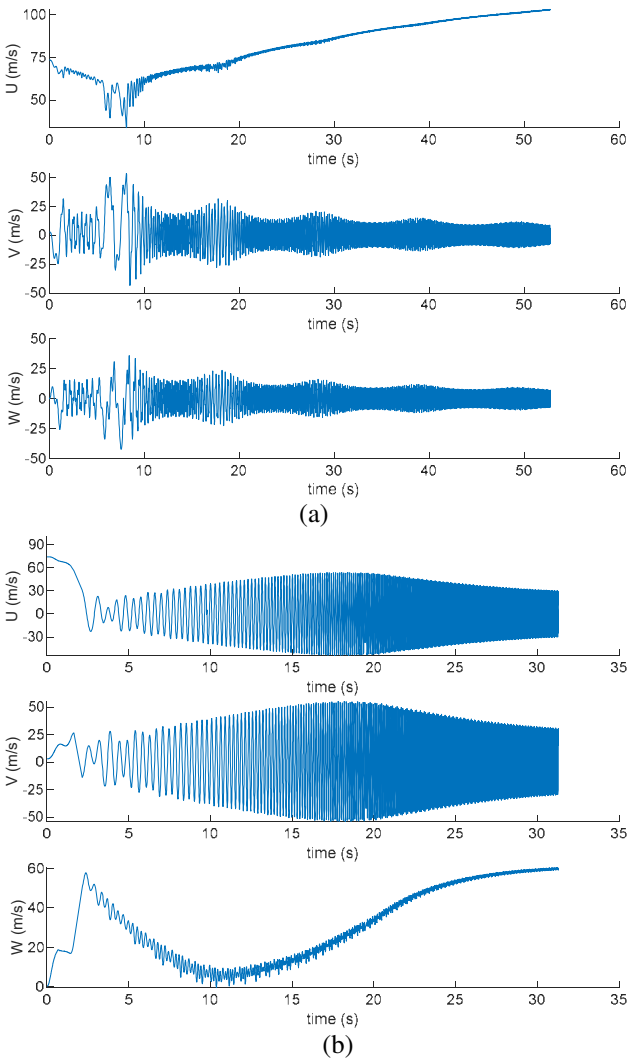

Fig. 2. Velocity history from two trajectories from the same Monte Carlo simulation with similar initial speeds and the same design parameters and initial height.

(a) Initial Speed: 73.7 m/s, Final Kinetic Energy at time of crash: 5.65 Megajoules

(b) Initial Speed: 74.1 m/s, Final Kinetic Energy at time of crash: 2.37 Megajoules

Figure 3 and Table VI describe several of the CPDs that we generated and their associated max kinetic energy at time of crash for different sets of design and operating parameters. Figure 3 shows several expected trends: failures that occur at higher altitudes lead to a larger and more spread out CPD and higher speeds moved the centroid of the CPD forwards (a, b) and caused a more spread out CPD (a, b, d, f). The roll of the UAS produced a bias towards the direction of the banked turn that the roll would cause (see d). Negative pitch angles (b, c) shifted the CPD's centroid away from the point where the UAS starts crashing, while positive and near horizontal pitch angles (a, e) moved the centroid the other way. Low masses increased the size of the CPD (e, f), which blurred some of the trends present in CPDs with higher mass (a, b, c, d). The CPDs with the lowest kinetic energy at the time of crashing (c, e) correlated to either a low height before crashing (c) or a low mass (e). Apart from mass's already known effect on kinetic energy, the height, speed, and wingspan appear to affect the UAS's kinetic energy when it crashes.

However, the results presented in Figures 2 and 3 and Table VI are only a snapshot of the 250 Monte Carlo simulations run for different configurations of design and operating parameters. In order to draw more accurate conclusions about how these parameters affect the safety of a UAS, a statistical analysis is needed that can be used for analyzing the Monte Carlo simulation results.

V. STATISTICAL ANALYSIS OF MONTE CARLO RESULTS

Statistical tests were performed on the Monte Carlo results to verify that the parameters varied affected the CPD and the kinetic energy at time of crash of the example UAS.

A. *Kinetic energy at time of crash*

Because a UAS's kinetic energy at the time of crashing is a single value it was possible to test which parameters caused statistically significant effects on it by binning the data based on parameter values. For all possible pairs of input variables, analysis of variance (ANOVA) tests were conducted, to test for first and second order correlation effects between the parameters. Three equally spaced bins for each parameter were used to bin the maximum kinetic energy from each of the 250 Monte Carlo simulations. Table VII provides the p-values from these tests, entries between the same variable denote first order correlation effects being

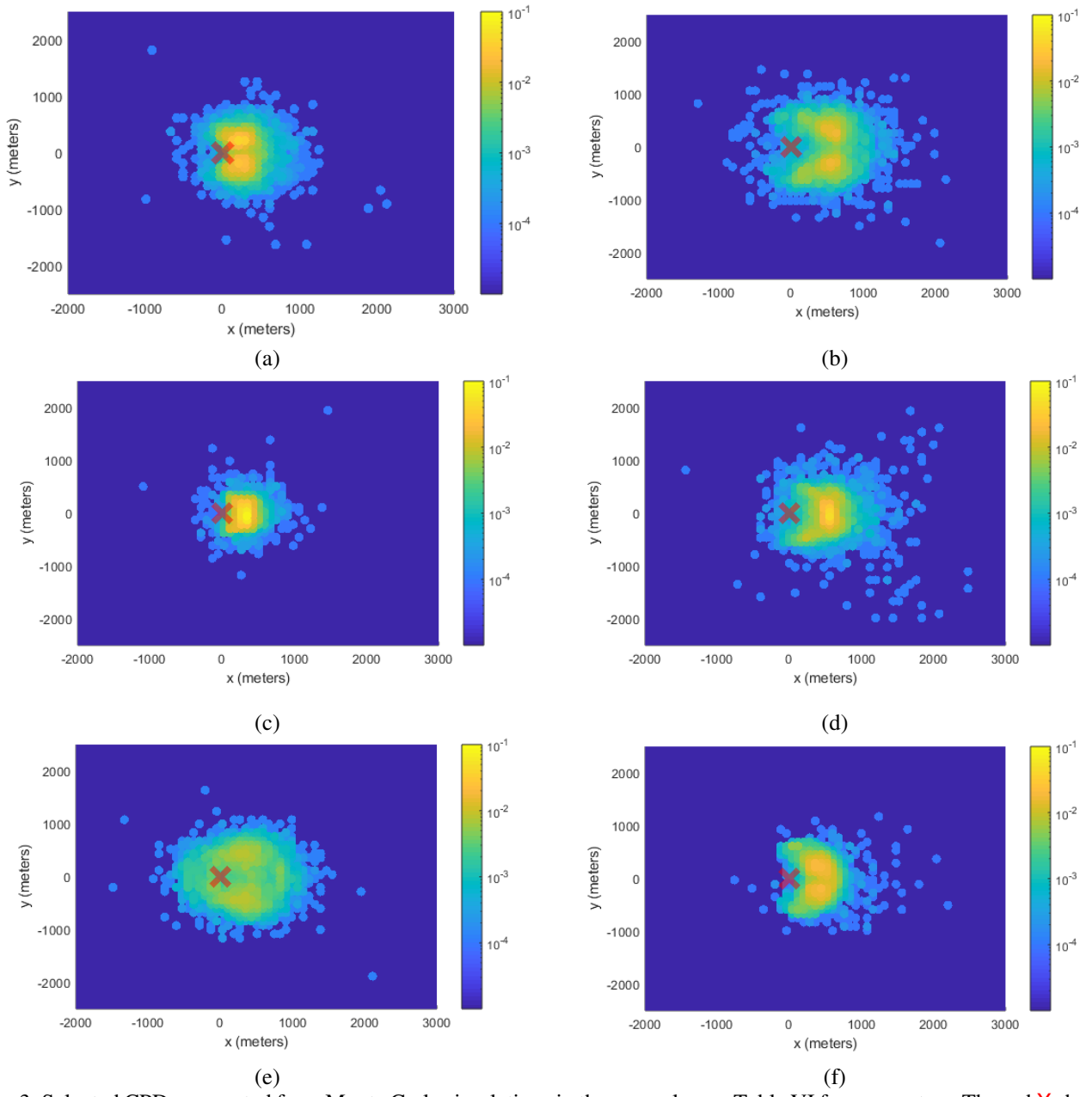

Fig. 3: Selected CPDs generated from Monte Carlo simulations in the example, see Table VI for parameters. The red X shows the point where the UAS begins crashing, the vehicle's velocity prior to crashing is in the same direction as the x-axis.

TABLE VI: Kinetic Energy at time of crash and parameters of selected example Monte Carlo simulations

Label	Kinetic Energy at time of crash (MJ)	Height (m)	Roll (deg.)	Pitch (deg.)	Speed (m/s)	Wingspan (m)	Rudder Range (deg.)	UAS Mass (kg)
(a)	15.6	1,898	26.8	0.89	37.5	15.1	36.0	1,356
(b)	27.0	1,969	-18.8	-8.84	60.6	12.9	33.5	1,378
(c)	8.54	658.7	-23.5	-9.25	41.7	16.8	31.6	1,166
(d)	14.5	871.2	-37.9	2.69	66.7	14.88	34.6	1,350
(e)	6.89	1,987	3.77	13.6	66.1	15.5	32.5	897.8
(f)	12.2	1,578	6.74	-1.82	42.3	7.52	37.5	995.0

present. The p-values in Table VII are the probability that the null hypothesis that there are no differences between the bins being compared is true. From the analysis in Table VII, we see that Height, Speed, Wingspan and Vehicle Mass can reject this null hypothesis (at a significance level of 0.05) and have statistically significant effects on the kinetic energy of the example UAS. Analytically this makes sense, height, speed and mass affect the initial potential and kinetic energy of the UAS before it begins crashing and wingspan affects the lift and drag on the UAS while it crashes.

B. Crash Probability Distribution

Because a CPD is a probability distribution, we analyzed the CPDs that we generated by calculating the sample means, covariances, coskewness [23] and cokurtosis [23]. The sample mean provides a statistic of where the centroid of the distribution is. The sample covariance provides a statistic describing the width and shape of the distribution. The sample coskewness provides a statistic describing the directional bias of the crash distribution. The sample cokurtosis provides a statistic capturing how spread out the distribution is. These four multivariate statistics were computed for each of the 250 CPDs generated. Note that the sample cokurtosis and coskewness varied noticeably between the 250 crash distributions, meaning that it would be inaccurate to use multivariate normal distributions to approximate the CPDs.

For all possible pairs of input variables, multiple analysis of variance (MANOVA) tests were conducted, in order to test for first and second order correlation effects between the input variables. The tests were conducted with a p-value of 5%, three equally spaced bins for each input variable were used and the output variables were the statistics computed for each of the 250 crash distributions. The results from these tests showed that there were statistically significantly differences between all possible pairs of input variables, indicating that there were statistically significant differences in the CPDs caused by the parameters considered. However, this analysis provides no information about how these parameters affect the CPDs.

A correlation analysis was performed to gain a basic understanding of how the input parameters affect the shape of the crash distribution. Correlation coefficients [24] were computed for the input variables relative to the computed statistics, using MATLAB's[22] corrcoeff function. The computed correlation coefficients were tested against the null hypothesis of the coefficients occurring due to random chance, using the corrcoeff function [22]. The probabilities of a correlation being present were identified between the input variables and the statistics computed for crash distributions, which are displayed in Table VIII.

From the parameters affecting the mean we see that the centroid of the CPD in the direction the UAS travels in is affected by pitch, speed, wingspan and mass. The shift in the CPD's centroid perpendicular to this direction is affected by roll and wingspan.

From the parameters affecting the variance statistics, we see that the size of the CPD is affected by height, pitch and speed, though pitch only affects the width of the CPD. Additionally roll affects the orientation of the CPD.

From the parameters affecting skewness, we see that the overall bias of the CPD in the direction the UAS travels in is affected by height, wingspan and mass. However, the bias perpendicular to this direction is unaffected by the parameters considered in this study. Speed, wingspan and mass affect the directionality of the bias of the CPD.

From the parameters affecting kurtosis we see that similar parameters affect the spread of the CPD as those that affect the size of the CPD. However, rudder range affects the spread of the CPD, but not its size.

Although the statistics in Tables VII and VIII are useful for understanding which aspects of a UAS's CPD and kinetic energy at time of crashing are affected by parameters, they do not allow for actually predicting the CPD of the UAS for a given set of parameters. Additionally, they are computed using a large number of computationally expensive Monte Carlo simulations. Having a CPD model that can be evaluated quickly is important when assessing safety during UAS operations or when UAS safety is being maximized using a numerical optimization method. To enable this type of analysis, we used the proposed approach to construct surrogate models for the CPD and kinetic energy of a UAS that could be evaluated quickly.

VI. SURROGATE MODELING UAS SAFETY USING A K-NEAREST NEIGHBORS (KNN) MODEL

A. Testing UAS surrogate models

In order to test the surrogate models developed, a test set of 50 random sets of parameters were drawn from a uniform distribution and the Monte Carlo simulations were run for these parameters. The weights, number of neighbors and inverse weighting exponent for the KNN were determined empirically without considering the test data set. The test set was not used as input data to the KNN, thus it could be used to assess the accuracy of the KNN model by comparing the results from the KNN model against those in the test set.

B. Modeling kinetic energy at time of crashing

The statistical analysis of Section V showed that the kinetic energy at time of crashing for the example UAS model was only affected by height, speed, wingspan and mass. Thus $w_{roll} = w_{pitch} = w_{rudder} = 0$. Additionally, we know part of the relation between mass and the kinetic energy at time of crashing ($1/2\,mv^2$), thus we use the KNN model to

TABLE VII: Statistical significance (p- values) of combinations of parameters affecting the UAS's kinetic energy at time of crashing, computed using ANOVA. Correlations between a parameter and itself indicate the parameter has a statistically significant effect when considered on its own. Bolded p-values have statistically significant effects on kinetic energy.

	Height	Roll	Pitch	Speed	Wingspan	Rudder Range	Vehicle Mass
Height	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Roll	**0.00**	0.57	0.24	0.08	**0.02**	0.58	**0.00**
Pitch	**0.00**	0.24	0.08	**0.01**	**0.00**	0.29	**0.00**
Speed	**0.00**	0.08	**0.01**	**0.00**	**0.00**	**0.01**	**0.00**
Wingspan	**0.00**	**0.02**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Rudder Range	**0.00**	0.58	0.29	**0.01**	**0.00**	0.15	**0.00**
Mass	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

TABLE VIII: p-values for null hypothesis that no correlation exists between input variables and distribution statistics. Bolded p-values are statistically significant (rejecting the null hypothesis), indicating that the input variable affects the distribution statistic in question.

Statistic	Mean		Covariance			Coskewness				Cokurtosis				
Variable	X	Y	X	Y	XY	X	Y	XXY	XYY	X	Y	XXXY	XXYY	XYYY
Height	0.33	0.70	**0.00**	**0.00**	0.79	**0.00**	0.74	0.27	0.20	**0.00**	**0.00**	0.97	**0.00**	0.89
Roll	0.06	**0.00**	0.62	0.53	**0.00**	0.21	0.31	0.34	0.67	**0.03**	0.75	**0.01**	0.40	**0.02**
Pitch	**0.00**	0.83	1.00	**0.02**	0.67	0.66	0.45	0.98	0.44	0.08	**0.00**	0.20	**0.00**	0.31
Speed	**0.00**	0.34	**0.00**	**0.00**	0.31	0.72	0.41	0.35	**0.00**	**0.00**	0.13	0.24	**0.01**	0.15
Wing-span	**0.00**	**0.03**	**0.00**	0.09	0.23	**0.00**	0.08	**0.02**	**0.00**	**0.01**	**0.00**	0.62	0.61	0.72
Rudder Range	0.09	0.46	0.32	0.70	1.00	0.32	0.57	0.21	0.46	**0.00**	**0.00**	0.38	**0.00**	0.82
Mass	**0.00**	0.72	**0.00**	0.27	0.15	**0.04**	0.11	0.09	**0.05**	0.07	0.11	0.71	0.20	0.61

approximate the speed (v^2) of the UAS at the time of crashing and then compute the kinetic energy as $1/2\,mv^2$.

Empirically, it was found that $w_{height} = 0.71$, $w_{speed} = 0.24$, $w_{wingspan} = 0.99$, $w_{mass} = 0.86$, $u = 1.7$, $k = 13$ yielded an acceptable KNN model. With these parameters, the KNN had a root mean squared error (RMSE) of 2.0 Megajoules on the test dataset, with a maximum error magnitude of 4.15 Megajoules. The highest maximum kinetic energy at time of crashing in the test data set was 26 Megajoules, the lowest maximum was 5.6 Megajoules.

C. *Modeling the UAS's CPD*

Because all parameters showed evidence earlier of affecting the CPD of the example UAS, they were all considered in the KNN model for the CPD. When modeling the CPD, we separate the CPD into a 2-D binned frequency distribution of a fixed size and the centroid of the distribution. This made it easier for the KNN to model effects that changed the shape of the CPD.

Empirically, it was found that $w_{height} = 0.2$, $w_{roll} = 0.76$, $w_{pitch} = 0.34$, $w_{speed} = 0.97$, $w_{wingspan} = 0.77$, $w_{rudder} = 0.19$, $w_{mass} = 0.17$, $u = 1.50$, $k = 2$ yielded an acceptable KNN model. Table IX details the RMSE and maximum error observed in each of the statistics discussed earlier between the CPDs produced by the KNN and those computed from the results of the test set.

D. *Computational Time*

The KNN models for both kinetic energy and the CPD required no more than 2 milliseconds to generate a result. In comparison the times needed to run one of the Monte Carlo simulations to generate a CPD in the test data set ranged from 1.8 to 32 hours.

E. *Discussion*

Figure 4 shows a comparison of several CPDs in the test set against the CPDs produced by the KNN model for the same parameters (Table X). The CPD KNN model's performance is more difficult to quantify, while its RMSE is low for the statistics considered, the maximum errors for several statistics are significantly higher. Part of this can be attributed to the KNN model's tendency to approximate larger CPDs than the CPD produced via Monte Carlo simulation, as seen in Figure 4b. Additionally, the KNN model sometimes misses behavior that may not have been present in the CPDs from the parameter study, which the KNN uses, such as the behavior in Figure 4c. However, the KNN also obtains slightly smoother CPDs that the Monte Carlo simulations, as seen in Figure 4b. This can lead to behavior where the KNN model misses a part of the CPD, like in Figure 4a.

However, the KNN model still produces CPDs that are visually close to the more accurate CPDs produced using Monte Carlo simulation, while using six orders of magnitude less computational time than a Monte Carlo simulation.

VII. CONCLUSIONS

This paper presented an approach for assessing UAS safety metrics by using Monte Carlo simulation to simulate the UAS crashing. A surrogate modeling approach was also presented, which can use the results of the Monte Carlo simulations to rapidly estimate UAS safety metrics for

TABLE IX: Quality of fit of KNN CPD model in terms of scaled root mean square error (RMSE) and scale maximum magnitude of error (Max Err.). Values scaled based off range of statistics present in the Monte Carlo simulations from Section V

Statistic	Mean		Covariance			Coskewness				Cokurtosis				
	X	Y	X	Y	XY	X	Y	XXY	XYY	X	Y	XXXY	XXYY	XYYY
RMSE	0.11	0.10	0.18	0.19	0.10	0.12	0.15	0.12	0.15	0.13	0.19	0.08	0.15	0.1
Max Err.	0.27	0.3	0.42	1.1	0.45	0.40	0.45	0.38	0.34	0.56	0.76	0.21	0.67	0.42

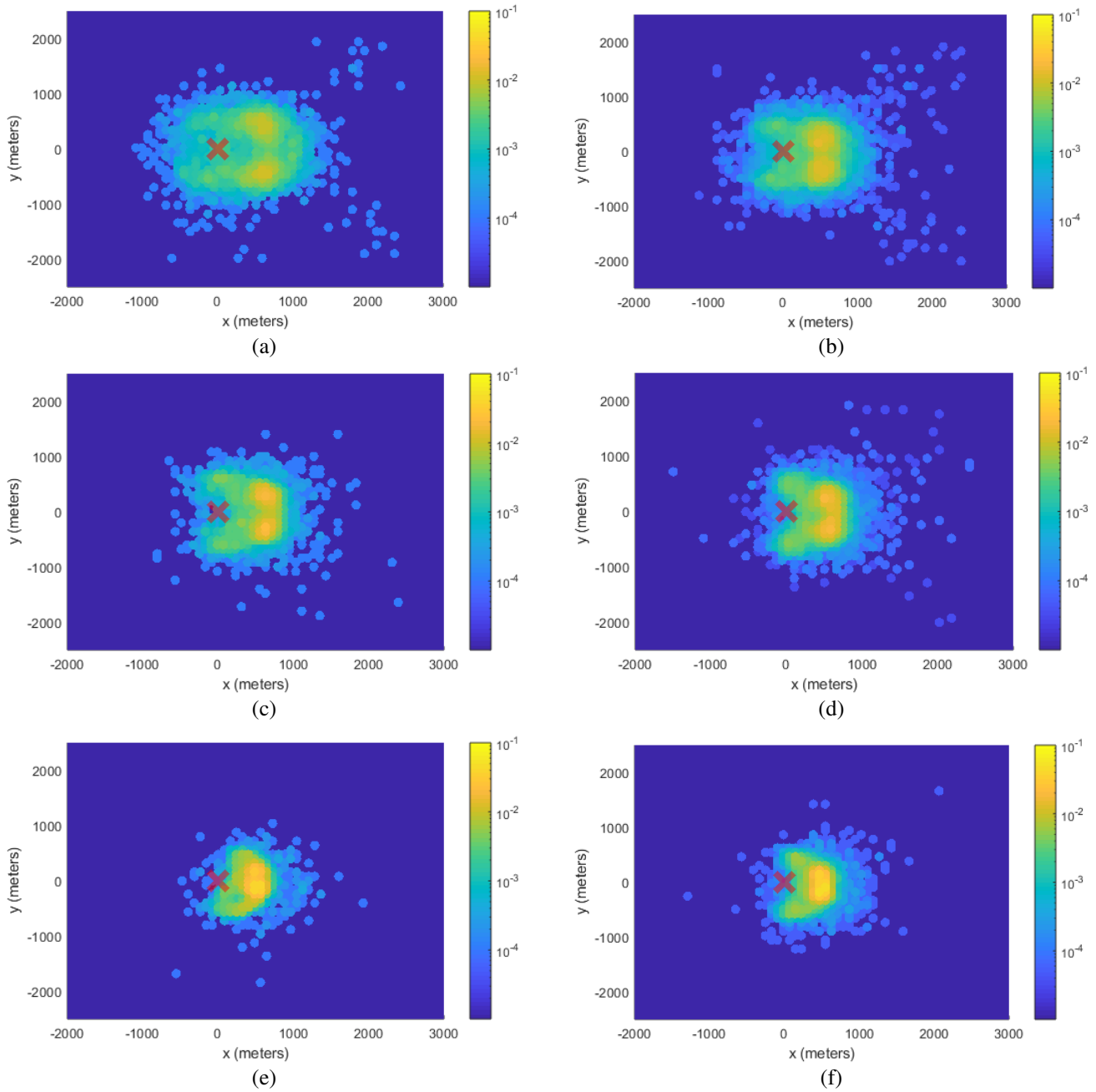

Fig. 4: Left (a, c, e): CPDs produced by the KNN model. Right (b, d, f): CPDs produced via Monte Carlo simulation

TABLE X: parameters of selected examples from test set used for comparison

Label	Height (m)	Roll (deg.)	Pitch (deg.)	Speed (m/s)	Wingspan (m)	Rudder Range (deg.)	UAS Mass (kg)
(a)	1,806	-19.3	1.83	69.8	12.2	36.0	898.2
(b)	1,307	-0.48	4.97	63.7	8.7	37.6	928.4
(c)	872.5	-40.5	-13.4	52.6	7.56	38.5	891.3

different UAS parameters. These approaches were demonstrated on an example UAS model based off a Cessna 182 in order to estimate two safety metrics for different design and operating parameters, the kinetic energy when impacting the ground and the UAS's CPD. A statistical analysis was conducted to assess the effects of these parameters on the two safety metrics. Results were presented comparing the results from the surrogate modeling approach against the Monte Carlo simulation approach for the two safety metrics considered.

The results indicate that the proposed approach was more accurate at modeling kinetic energy than it was at modeling CPDs. This can be partially attributed to the fact that the CPD KNN model used a k value of 2, which generates CPDs using only two reference CPDs. However, the performance of the CPD KNN could not be improved by increasing the k value.

This indicates that there may more effective surrogate approaches for estimating CPDs than a KNN model, or that increasing the number of experiments in the DOE would have been beneficial. The k value could have been set higher had a Full Factorial DOE been used instead of a Latin Hypercube DOE. However, a Full Factorial DOE requires an exponential number of experiments relative to the number of parameters being varied, which is only computationally feasible when considering a small number of parameters.

The KNN models were several orders of magnitude faster than the Monte Carlo simulations, which is an acceptable tradeoff for the loss in accuracy relative to them. Critically, the KNN models are fast enough that they could be potentially evaluated in real time. Such models can facilitate UAS performing online risk management and make it computationally feasible to optimize UAS safety. Thus, the proposed modeling approach shows promise for enabling new approaches to managing UAS safety.

VIII. REFERENCES

[1] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*. DARcorporation, 1998.

[2] A. Washington, R. A. Clothier, and J. Silva, "A review of unmanned aircraft system ground risk models," *Progress in Aerospace Sciences*, vol. 95, pp. 24–44, 2017.

[3] R. Clothier, B. Williams, and K. Hayhurst, "Modelling the risks remotely piloted aircraft pose to people on the ground," *Safety Science*, 2017.

[4] D. A. Burke, "System level airworthiness tool: A comprehensive approach to small unmanned aircraft system airworthiness." Ph.D. dissertation, North Carolina State University, 2010.

[5] E. Rudnick-Cohen, J. W. Herrmann, and S. Azarm, "Risk-based path planning optimization methods for unmanned aerial vehicles over inhabited areas," *Journal of Computing and Information Science in Engineering*, vol. 16, no. 2, p. 021004, 2016.

[6] A. la Cour-Harbo, "Quantifying risk of ground impact fatalities of power line inspection bv los flight with small unmanned aircraft," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 1352–1360.

[7] S. Primatesta, A. Rizzo, and A. la Cour-Harbo, "Ground risk map for unmanned aircraft in urban environments," 2018, preprint. [Online]. Available http://www.forskningsdatabasen.dk/en/catalog/2436455221.

[8] A. Ford and K. McEntee, "Assessment of the risk to ground population due to an unmanned aircraft in-flight failure," in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2010, p. 9056.

[9] P. P. Wu and R. A. Clothier, "The development of ground impact models for the analysis of the risks associated with unmanned aircraft operations over inhabited areas," in *Proceedings of the 11th probabilistic safety assessment and management conference (PSAM11) and the annual European safety and reliability conference (ESREL 2012)*, 2012.

[10] J. Breunig and S. Sayed, "Modeling ground collision severity of small unmanned aircraft systems," in *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3349.

[11] R. Aalmoes, Y. Cheung, E. Sunil, J. Hoekstra, and F. Bussink, "A conceptual third party risk model for personal and unmanned aerial vehicles," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 1301–1309.

[12] C. W. Lum and D. A. Tsukada, "UAS reliability and risk analysis," in *Encyclopedia of Aerospace Engineering*, Chichester, West Sussex, U.K.: Wiley, 2016 pp. 1–12.

[13] A. la Cour-Harbo, "Quantifying risk of ground impact fatalities for small unmanned aircraft," *Journal of Intelligent & Robotic Systems*, pp. 1–18, 2018.

[14] A. la Cour-Harbo and H. Schioler, "Ground impact probability distribution for small unmanned aircraft in ballistic descent," 2017, preprint. [Online]. Available: http://vbn.aau.dk/da/publications/ground-impact-probability-distribution-for-small-unmanned-aircraft-in-ballistic-descent(dc2d6dc0-30f4-44cd-a989-8962b925a76).html

[15] C. Lum, K. Gauksheim, C. Deseure, J. Vagners, and T. McGeer, "Assessing and estimating risk of operating unmanned aerial systems in populated areas," in *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIAA Balloon Systems Conference and 19th AIAA Lighter-Than*, 2011, p. 6918.

[16] E. S. Rudnick-Cohen, S. Azarm, and J. Herrmann, "Multi-objective design and path planning optimization of unmanned aerial vehicles (UAVs)," in *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015, p. 2322. [Online]. Available: http://arc.aiaa.org/doi/-pdf/10.2514/6.2015-2322

[17] Y. Haartsen, R. Aalmoes, and Y. Cheung, "Simulation of unmanned aerial vehicles in the determination of accident locations," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. IEEE, 2016, pp. 993–1002.

[18] United States Federal Aviation Administration, "Draft-NPRM operation of small unmanned aircraft systems over people," Proposed changes to 14 CFR part 107, Jan. 2019, [Online]. https://www.faa.gov/uas/programs_partnerships/DOT_initiatives/media/2120-AK85_NPRM_Operations_of_Small_UAS_Over_People.pdf.

[19] Range Commanders Council, "Common risk criteria for national test ranges (supplement)", 321-07, U.S. Army White Sands Missile Range, New Mexico, 2007.

[20] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE suite," *SIAM journal on scientific computing*, vol. 18, no. 1, pp. 1–22, 1997.

[21] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*. ACM, 1968, pp. 517–524.

[22] "MATLAB," 2018a, the MathWorks, Natick, MA, USA.

[23] M. B. Miller, *Mathematics and Statistics for Financial Risk Management*. John Wiley & Sons, 2013.

[24] W. Härdle and L. Simar, *Applied Multivariate Statistical Analysis*. Springer, 2007, vol. 22007.