A Utility-Based Path Planning for Safe UAS Operations with a Task-Level Decision-Making Capability

Uluhan C. Kaya[1], Atilla Dogan[2], and Manfred Huber[3]

Abstract— **Unmanned aircraft systems (UAS) are being used more and more every day in almost any area to solve challenging real-life problems. Increased autonomy and advancements in low-cost high-computing technologies made these compact autonomous solutions accessible to any party with ease. However, this ease of use brings its own challenges that need to be addressed. In an autonomous flight scenario over a public space, an autonomous operation plan has to consider the public safety and regulations as well as the task specific objectives. In this work, we propose a generic utility function for the path planning of UAS operations that includes the benefits of accomplishing the goals as well as the safety risks incurred along the flight trajectories, with the purpose of making task-level decisions through the optimization of the carefully constructed utility function for a given scenario. As an optimizer, we benefited from a multi-tree variant of the optimal T-RRT* (Multi-T-RRT*) path planning algorithm. To illustrate its operation, results of simulation of a UAS scenario are presented.**

I. INTRODUCTION

Recent advancements on the unmanned systems manifest the potential of these technologies to impact our daily life. In particular, the unmanned aircraft systems (UAS) become ordinary for people in almost any area from aerial photography to emergency responses, from agricultural services to even autonomous deliveries. Easiness and affordability to access these systems accelerated the innovations and the novel ideas for the solution of diverse real-life problems. Despite its benefits, however, this widespread availability also resulted in the safety and regulatory concerns. In an autonomous flight task over a public space, besides the mission objectives, concerns regarding the public safety, privacy, and the regulations have to be addressed during the planning and considered in the decision-making process. Therefore, there is a need for the multi-objective decision-making capability in a path planning process that can also quantify and compare the mission objectives and the risks incurred by the mission.

Although plenty of studies have tackled the path planning of the autonomous aircraft, just a few approaches to address the concerns mentioned above. Various risk assessment frameworks have been introduced to quantify the risk of UAS operations over the national airspace. Ancel et al. use the probabilistic real-time component failure analysis in their framework to assess the risk to the people on the ground in [1]. Also, the operational risk of multiple UAVs in a partially unknown environment is minimized by the multi-criteria path optimization as the trade-off between the path-integral risk measure and the distance in [2] using Gaussian process-based stochastic environmental factors. On the other hand, the majority of the studies attempts to solve the path planning of UAS considering simple cost considerations such as total distance or mechanical work but using computationally more efficient algorithms. Despite its simplicity, one of the most practiced and powerful algorithms among these is rapidly-exploring random trees (RRT). RRTs are used in various challenging planning tasks and achieved significant progress. [3] utilizes the multi-tree variant of an optimal T-RRT* algorithm for the task ordering and the pathfinding in continuous cost space. Also, a minimum risk-based variant of an optimal RRT* with a multi-criteria objective function is introduced in [4] using user-defined primary and the secondary objective functions.

In this study, we aimed to benefit from the concept of utility to construct a generic function for the path planning purpose such that the safety related concerns and the mission objectives of the UAS operations can be expressed within the same function and also, the optimization of such a function can be used to make task-level decisions. Proposed utility function quantifies the profits of the mission objectives and the risks incurred along the mission as a path integral which can be efficiently optimized by current RRT algorithms. For the optimization of the utility along the trajectories, a slightly modified multi-T-RRT* algorithm will be used.

This paper is organized as follows. In Section II, the formulation of the path utility function and the node utilities is explained. Section III describes the multi-T-RRT* algorithm over the pseudo code. Simulation results and the implementation details of the scenarios are given in IV, and finally, the conclusions are drawn.

II. CONSTRUCTION OF THE UTILITY FUNCTION

A. Path Utility Function

Path utility is defined as the cumulative utility obtained along the trajectory of a UAS mission and is formulated with a path integral in (1).

$$\mathcal{U}(\tau_{0..T}) = \int_0^T P(Success|\tau_{0..t})\lambda_{\mathcal{U}}(\tau_t|\tau_{0..t})\,dt \qquad (1)$$

This work was supported by National Science Foundation

[1]Uluhan Kaya is with the Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, Arlington, TX 76019, USA uluhan.kaya@mavs.uta.edu

[2]Atilla Dogan is with the Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, Arlington, TX 76019, USA

[3]Manfred Huber is with the Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington, TX 76019, USA

The path integral is composed of two key factors. The first one is the success probability of the platform to reach the current point τ_t along its planned trajectory $\tau_{0..t}$ and the other factor is the expected rate of reward/payoff, $\lambda_{\mathcal{U}}$, at the point given the current trajectory. In this integration, the first term plays the role of a weighting factor for the utility contributions of the point to the cumulative path utility based on the likelihood that this point will actually be reached along the trajectory. This weighting also implicitly states that if the vehicle does not succeed to reach a point, either due to a catastrophic failure or the termination of the mission, the utilities of the rest of the trajectory should not be considered in the path utility.

The success probability of the platform depends on numerous factors in a real UAS scenario which can be analyzed using many different approaches. In this work, we use a general terminology of events that can happen on or around the platform, and once happened, can impact the success of the platform. We categorized these events as catastrophic events and non-catastrophic events. In the case of a catastrophic event, the platform undergoes an uncontrollable regime, resulting in the termination of the mission with a potential impact to the ground. In a non-catastrophic event, on the other hand, the platform is assumed to be controllable and can still continue its mission. While other events, such as partially controllable failures can occur in the real world, and can be integrated into the framework, we will here concentrate on only these two event types, resulting in the situation where only the catastrophic events can impact the success of the platform by causing its complete failure. Under this assumption, the success probability of the platform to reach the current trajectory point is equal to the probability of not having encountered any catastrophic failures, F_c, until this point. Using this fact, (1) becomes (2)

$$\mathcal{U}(\tau_{0..T}) = \int_0^T P(\neg F_c|\tau_{0..t})\lambda_{\mathcal{U}}(\tau_t|\tau_{0..t})\,dt \tag{2}$$

The reward rate $\lambda_{\mathcal{U}}$ represents the expected incremental utility contribution of the events happening at trajectory location τ_t, and can be defined for the two categories of events considered here, which are events that can only happen once or that can happen multiple times over the trajectory. For the former category, the underlying reward from an event can be obtained only at a given point and on following occurrences to the same point, the same event does no longer contribute to the path utility. This type of events may model the utility of reaching an intermediate goal location, or having a permanent component failure. The reward rate, $\lambda_{\mathcal{U}}$, for these one-time events can be expressed as follows:

$$\lambda_{\mathcal{U}}(\tau_t|\tau_{0..t}) = \sum_{i \in Events} P(\neg e_i|\tau_{0..t})\lambda_{e_i}(\tau_t|\tau_{0..t})\mathcal{U}_i(\tau_t|\tau_{0..t})w_i \tag{3}$$

where $\tau_{t_{e_i}}$ is the trajectory node at which the event e_i happens, and w_i is the relative weighting factor of the utility contribution of the i^{th} event at the same node. It is important to note here that for catastrophic events, the term $P(\neg e_i|\tau_{0..t})$ moves out of the sum and becomes part of the probability that the trajectory is still active, $P(\neg F_c|\tau_{0..t})$, in (2) since occurrence of a catastrophic event terminates any future utility of any other event.

The reward rate for events that may happen multiple times at a trajectory location are modeled with a rate parameter of the event, defined in unit time, as below. Non-critical, temporary platform or component failure events, as well as general costs incurred along the trajectory can be modeled by this approach.

$$\lambda_{\mathcal{U}}(\tau_t|\tau_{0..t}) = \sum_{i \in Events} \lambda_{e_i}(\tau_t|\tau_{0..t})\mathcal{U}_i(\tau_t|\tau_{0..t})w_i \tag{4}$$

where $\lambda_{e_i}(\tau_t|\tau_{0..t})$ is the expected number of times per time unit that event e_i is happening at point τ_t given the trajectory.

In the case of events that are guaranteed to occur at a particular point along the trajectory such as, for example, the reaching of a delivery location or events caused directly by control actions, the event rate, $\lambda_{e_i}(\tau_t|\tau_{0..t})$, can be expressed (with a slight abuse of notation) by a Dirac delta function as $\lambda_{e_i}(\tau_t|\tau_{0..t}) = \delta(\tau_t - \tau_{t_{e_i}}|\tau_{0..t})$. This allows the incorporation of mission goals associated with specific actions or locations.

For a mission having a mixture of these events, the expected reward rate term in (2) becomes the sum of (3) and (4) with corresponding events.

With this approach we propose to facilitate a task-level decision-making capability through the maximization of a carefully constructed path utility function that includes possible task centric events and utility components such as reaching a destination, accomplishing a task component, violating safety or regulatory considerations, or failure of the platform. According to the objectives of an assigned task and the other considerations like ground safety, the proposed utility approach allows to consider and generate solutions other than purely completing the mission. An example for this could be a "stay on the ground" decision of a UAS, which might yield the highest utility, for a package delivery mission in adversarial weather conditions. With its bare interpretation, in the proposed framework, the final decision is to choose the highest utility path which can be extended to highly complex scenarios with a detailed analysis of the events that can happen during a UAS mission.

B. Calculation of Point Utilities

Rewards/Utility changes obtained at a specific point are calculated according to the events happening on the platform that affect the ground or the mission parameters. These events are defined to capture the mission objectives and the safety concerns. For an illustration, the event of taking aerial pictures of an area at a trajectory point has an effect on the mission and on the ground, which yields the utility of that point. In this case, for example, taking pictures yields a benefit in terms of obtaining the pictures but also incurs a cost in terms of the potential invasion of privacy of persons on the ground. Another example is that the event of a catastrophic failure on the platform at a trajectory point has also an impact on the ground in terms of potential injury and property damage caused by the impact on the ground with an associated utility.

Calculation of the node utility can be divided in two stages. The first stage is to find the spatial distribution of the attainable utility at a location on the ground given the previous trajectory and the point at which the event happened. In the second stage, the utilities for ground locations are integrated over the attainable area impacted by the event.

The first stage of the calculation is shown in (5). This term is the spatial map of the utility for the given event.

$$\mathcal{U}_{i,\tau,t}(\mathcal{X}) = p(\mathcal{X}|\tau_{0..t}, \tau_{t_{e_i}})\mathcal{M}(\mathcal{X}|\tau_{0..t}, \tau_{t_{e_i}}) \tag{5}$$

Here $\mathcal{X}$ is the location on the ground, and $\mathcal{U}_{i,\tau,t}(\mathcal{X}) = \mathcal{U}_i(\mathcal{X}|\tau_{0..t}, \tau_t = \tau_{t_{e_i}})$.

In the second stage of the point utility calculation, the expected utility increment is computed by integration of the location-dependent utilities on the ground weighed by the probability distribution of the attainability of those locations for an event, e_i, happening at the current trajectory point, is carried out over the attainable area A_{e_i}:

$$\mathcal{U}_i(\tau_t|\tau_{0..t}) = \int_{A_{e_i}} p(\mathcal{X}|\tau_{0..t}, \tau_{t_{e_i}})\mathcal{U}_{i,\tau,t}(\mathcal{X})\, d\mathcal{X} \tag{6}$$

$p(\mathcal{X}|\tau_{0..t}, \tau_{t_{e_i}})$ here is the probability distribution of attainability of the locations, $\mathcal{X}$, on the ground for an event happening at the current trajectory point, and it represents the likelihood of obtaining the underlying utility at each location. It is important to note here that the attainability of a ground location concept is used in a wide sense here. For some events such as catastrophic failures, it can be defined as the impact probability at a location, or for the event of taking aerial pictures of an area, it could be represented as the success probability of covering a location on the ground in the picture.

The steps for the construction of (5) and the integration of (6) is described in more detail in [5] for the risk of exposure of the ground to the failing UAS platform. In this previous work it was assumed that the previous trajectory does not affect the risk of the current point and the underlying risk exposure distribution is static. Under this assumption, the risk associated with the trajectory point where the failure happened is given in the equation below:

$$R_F(\tau_t) = \int_{A_F} p(\mathcal{X}_{imp}|\tau_t = \tau_{t_F})PREM(\mathcal{X}_{imp})\, d\mathcal{X}_{imp} \tag{7}$$

where the first term in the integral is the UAS impact location distribution on the ground given the trajectory node of failure and $PREM$ (Probabilistic Risk Exposure Map) is the distribution of the risk of exposure on the ground.

The plot on the left in Fig 1 illustrates the $PREM$ and the UAS impact location on the ground. In the right plot, the spatial distribution of the impact utility (risk) is sketched. Note that the formulation of proposed utility function is flexible enough to accommodate most commonly used cost functions in path planning such as distance traveled or total time by treating them as an event happening every instance of time and assigning an appropriate utility. To illustrate this, time can be included using (4) by setting the first term (the event rate) to 1 and the utility term to -1, which integrates to the elapsed time along the trajectory as a negative utility component. Similarly, for the distance traveled case, one can assign the negative speed of the vehicle as the reward rate at a point to account for total distance traveled in the utility optimization of the path.

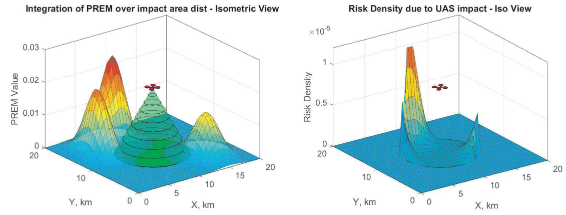


Fig. 1. Illustration of a failure event with $PREM$ and impact distribution (left) and the calculation of the risk of a point on the discritized failure area (right)

III. MULTI-T-RRT* PATH PLANNER

In our proposed utility based approach to the path and task planning problem of UAS operations, we are using a modified multi-tree variant of T-RRT* algorithms [3], [6], [7] as an optimization technique to maximize the proposed path utility function over the planned trajectories. The proposed algorithm grows a forward tree and multiple backward trees (one from each potential goal - i.e. allowed path end location) by iteratively sampling from the configuration space. The sampling uses the goal bias and transition test based heuristics to drive the extension of trees faster to the goal configurations and also to efficiently explore the high utility regions first. The pseudo-code of the algorithm is given in Algorithm 1. Provided with initial and goal configurations, goal connection biases of the forward and backward trees, and the connection threshold, the algorithm starts with initializing the forward tree T_{fw} from the initial configuration x_{init} and the backward trees T_{bw} from the goal configurations $\mathcal{X}_{goal}$, setting the path parameters to the empty set. In an iterative loop, first, the forward tree extends a branch towards a random configuration chosen from the $\mathcal{C}$ space, with the goal bias heuristic that samples also among $\mathcal{X}_{goal}$, if the extension passes the transition test shown in Algorithm 2. The transition test is a sample rejection heuristic which uses an adaptive probabilistic measure to eliminate the samples that require the extension of branches from low cost to high cost (or low utility) regions. Details of the implementation can be found in [3], [7]. If the test is passed, the extended node and the edge are added to the tree with its accumulate link utility. After that the near neighbor search function finds the neighbor nodes around q_{new} within a distance as in the RRT* algorithm [6]. The rewiring operation is performed between the near neighbors, Q_{near}, and the extended node q_{new}, and the tree is maintained.

Algorithm 1 Multi-T-RRT*

```
Input: $x_{init}, \mathcal{X}_{goal}, P_{goal}, P_{connect}, \mathcal{C}, thrs$
Output: $E_{path}, \mathcal{U}_{path}, \sigma_{path}, T_{fwd}, T_{bwd}$
  Initialize($x_{init}, \mathcal{X}_{goal}, T_{fw}, T_{bw}, E_{path}, \mathcal{U}_{path}, \sigma_{path}$)
for $i = 1$ to $iterMax$ do
    $q_{rand} \leftarrow ChooseTarget(\mathcal{X}_{goal}, P_{goal}, \mathcal{C})$
    $T_{fw} \leftarrow Extend\&Rewire\_Fwd(q_{rand}, T_{fw})$
    $[S, N_{IDs}, T_{IDs}] \leftarrow ConnectTrees(T_{fw}, T_{bw}, thrs)$
    if $S = True$ then
        $[E, \mathcal{U}, \sigma] \leftarrow CreatePath(N_{IDs}, T_{IDs}, T_{fw}, T_{bw})$
        if $\mathcal{U} > \mathcal{U}_{path}$ then
            $[E_{path}, \mathcal{U}_{path}, \sigma_{path}] \leftarrow [E, \mathcal{U}, \sigma]$
        end if
    end if
    $Q_{rand} \leftarrow ChooseTarget([x_{init}, q_{rand}], P_{connect}, \mathcal{C})$
    $T_{bw} \leftarrow Extend\&Rewire\_Bwd(Q_{rand}, T_{bw})$
    $[S, N_{IDs}, T_{IDs}] \leftarrow ConnectTrees(T_{fw}, T_{bw}, thrs)$
    if $S = True$ then
        $[E, \mathcal{U}, \sigma] \leftarrow CreatePath(N_{IDs}, T_{IDs}, T_{fw}, T_{bw})$
        if $\mathcal{U} > \mathcal{U}_{path}$ then
            $[E_{path}, \mathcal{U}_{path}, \sigma_{path}] \leftarrow [E, \mathcal{U}, \sigma]$
        end if
    end if
end for
return $E_{path}, \mathcal{U}_{path}, \sigma_{path}, T_{fw}, T_{bw}$
```

Algorithm 2 $Extend\&Rewire\_Fwd$

```
Input: $q_{rand}, T_{fw}$
Output: $T_{fwd}$
$q_{nearest} \leftarrow NearestNode(q_{rand}, T_{fw})$
$q_{new} \leftarrow steer(q_{rand}, q_{nearest})$
$\mathcal{U}(q_{new}) \leftarrow NodeUtility(q_{new})$
if $TransitionTest(T_{fw}, \mathcal{U}(q_{new}), \mathcal{U}(q_{nearest}))$ then
    $\mathcal{U}(\sigma_{new}) \leftarrow PathUtility\_Fwd(q_{new}, \sigma_{nearest})$
    $addNode(T_{fw}, [q_{new}, q_{nearest}], \mathcal{U}(q_{new}), \mathcal{U}(\sigma_{new}))$
    $Q_{near} \leftarrow NearNeighbors(T_{fw}, q_{new})$
    $T_{fw} \leftarrow Rewire\_Fwd(q_{new}, Q_{near}, T_{fw})$
end if
return $T_{fw}$
```

Algorithm 3 $CreatePath$

```
Input: $N_{IDs}, T_{IDs}, T_{fw}, T_{bw}$
Output: $E_{path}, \mathcal{U}_{path}, \sigma_{path}$
$T_{back} \leftarrow IdentifyBackwardTree(T_{IDs}, T_{bw})$
$[E, \mathcal{U}, \sigma]_{path} \leftarrow biBackTrack(T_{fw}, T_{back}, N_{IDs})$
return $E_{path}, \mathcal{U}_{path}, \sigma_{path}$
```

It should be noted here that some of the functions denoted with $\_Fwd$ or $\_Bwd$ are designated specifically for the forward or backward trees. The reason is that since the path utility function developed in (2) depends on the trajectory from root node τ_0 to current node and the backward trees have no information about the forward trajectory until they are connected, the calculation and the accumulation of path utilities are different for the forward and backward trees.

The $ConnectTrees$ function checks the possible connections between the forward tree and the backward trees, and if there are any, it returns with the IDs of the connected nodes and their corresponding tree identifiers, and with a boolean indicating the connection success. If the connection is successful, a path is created by backtracking from the connection nodes on the forward tree through the connected backward tree. According to the calculated utility of the path created, if it surpasses the best path found so far, the best path and its utility are updated to the new one.

The extension, rewiring, and the connection check processes are repeated for the backward trees individually in the same iteration, except with a slightly changed path utility calculation as mentioned above. If any higher utility path is found during the extension of the backward trees, the best path is again updated, and the whole iteration is repeated until the termination condition (in this case a maximum iteration count) is reached.

IV. SIMULATION RESULTS

For the demonstration of the proposed approach, we construct the path utility function for a simple UAS package delivery scenario where the events that can happen during the mission are catastrophic failure events, delivering the package to the designated location, landing on another warehouse/station to leave the package there or to land/stay at the take-off location. In this scenario, we model the catastrophic events on the vehicle as Poisson processes having constant failure rates using (4) and the other events as the single node events using (3). Node utilities for the corresponding events are presumed to depend only on the location of the current node and the underlying utility rate is independent of time or the previous trajectory. With these assumptions, the path utility function becomes as follows:

$$\mathcal{U}(\tau_{0..T}) = \int_0^T e^{-\lambda t} \lambda_{\mathcal{U}}(\tau_t)\, dt \tag{8}$$

One advantage of using Poisson process failure models is their independent and stationary memoryless properties. Although time is known for each node on the forward tree during extension, time of the nodes on the backward trees is relative to the unknown end time and it runs backward in time. Using the independent and stationary increments of the Poisson process, we can compute the relative increments of the path utilities over the backward trajectories and compute the complete path utility when connecting the forward to the backward branch using a simple operation. This also allows us to use rewiring operations efficiently on the backward trees.

Path utility propagation for the backward tree is formulated in (9). Trajectory nodes on a backward tree are given as $(\tau_T, ..., \tau_{t_{k+1}}, \tau_{t_k}, ...)$ starting from a goal node τ_T in backward order.

$$\mathcal{U}(\tau_{T..t_k}) = \int_{t_k}^{t_{k+1}} e^{-\lambda(t-t_k)}\lambda_{\mathcal{U}}(\tau_t)dt + e^{-\lambda\Delta t_k}\mathcal{U}(\tau_{T..t_{k+1}}) \tag{9}$$

where $\Delta t_k = t_{k+1} - t_k$ is the time increment while extending a backward branch. In this function, the integral term assumes that time starts from node τ_{t_k} and it computes the incremental utility of the path segment $(\tau_{t_k}, \tau_{t_{k+1}})$. The second term in the function adds the cumulative incremental utilities until the previous node $\tau_{t_{k+1}}$ by multiplying them with the incremental success probability during this extension.

Using the above functions, total path utility calculation when both trees are connected is given below.

$$\mathcal{U}(\tau_{0..T}) = \mathcal{U}(\tau_{0..t}) + e^{-\lambda t}\mathcal{U}(\tau_{T..t}) \tag{10}$$

Assuming both trees are connected at node τ_t, the first term computes the path utility accumulated by the forward tree extension until the connection, and the second term propagates the time on the backward tree nodes by adding the connection time t and the resulting utility accumulation over the backward tree, discounted by the likelihood that the connection point is reached.

A. Implementation Details

In this scenario, we selected a small region to demonstrate the proposed concept. The region, shown in Fig 2, consists of buildings around the roads. The mission is to deliver a package from a start location to a goal location with a highest possible path utility. The mission objectives are to deliver the package to the original destination with a minimum intrusion (risk) to the properties, and if the utility of delivering to this location is lower than the expected utility of the other delivery locations, then, consider delivering to the other locations, or do not take-off at all. These decisions are being made according to the comparison between the utility values of the mission objectives. The utilities are composed of the benefits obtained by completing the mission objectives as positive utility and the risk accumulated during the mission as negative utility.

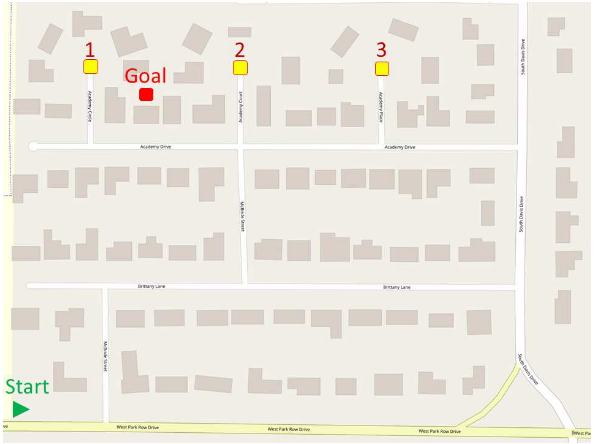


Fig. 2. UAS package delivery scenario map

In this scenario, the other package drop locations are shown on the map, and the utility of package delivery to these locations are determined according to their distance to the original drop location, where the numbers on the scenario map indicate the closeness of these locations to the goal. Point utility values of the mission objectives for all possible drop locations are selected as $10 \cdot 10^{-9}, 4 \cdot 10^{-9}, 3 \cdot 10^{-9}$ and $1 \cdot 10^{-9}$, respectively. In addition, 0 utility value is assigned to "Do-Not-Fly" action which means no benefits obtained. Note that in a real scenario, this might be a loss with a negative utility.

The same scenario is simulated under two different cases. First case represents the normal operating conditions with average failure rates, whereas, the second case stands for the condition with increased failure occurrences with the same impact regions. The occurrence rates of the selected failure events are given in Table I for both cases.

TABLE I
CATASTROPHIC FAILURE EVENT RATES

1/hr	F_1	F_2	F_3	F_4
Case 1	0.001	0.05	0.01	0.05
Case 2	0.01	0.5	0.1	0.5

Impact locations of the selected failure events are modeled using truncated Gaussian distributions such that 3σ standard deviation of the normal distribution fit into the elliptical regions, that define the impact domain of the event. Illustration of the regions are depicted in Fig 3.

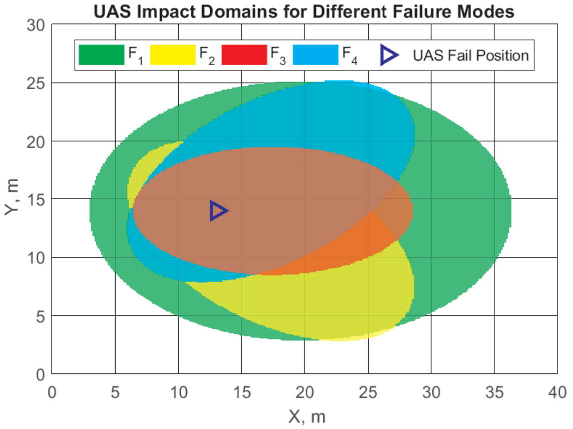
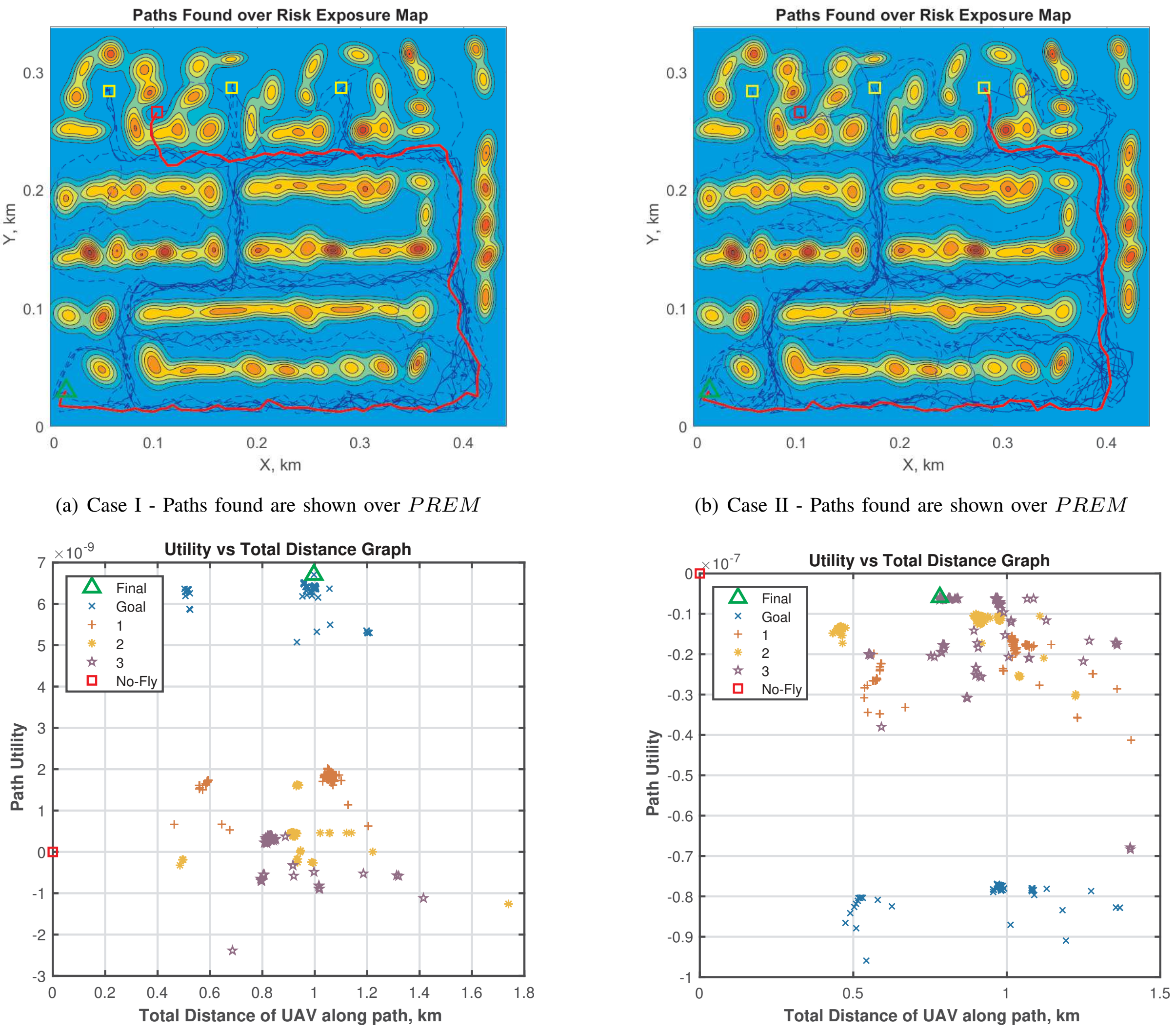


Fig. 3. Illustration of impact distributions of failure events on the ground

The aim is to show the effect of operating conditions on the task-level decision and the path characteristics.

B. Scenario Results

Paths that are found during the optimization of the proposed utility function are plotted over the risk exposure map of the area in Fig 4 for the selected cases. In these plots, the



(a) Case I - Paths found are shown over $PREM$

(b) Case II - Paths found are shown over $PREM$

(c) Case I - Utility vs Total Length of the Paths

(d) Case II - Utility vs Total Length of the Paths

Fig. 4. Simulation results for 2 cases

solid red line is the final path maximizing the utility, while the dashed blue lines are the other possible paths found with lower utilities. Also, the utilities of all the paths connecting to the possible goal locations are shown with respect to the total length of the paths found.

From these figures, it is seen that the optimization of the proposed utility function leads path searching to the less risky areas successfully before the mission completion. Intuitively, avoiding the narrow passages between buildings and flying over wider roads, where we omitted the traffic activity, would yield higher utilities when we consider the impact damage to the properties. However, behavior of the search can be affected by the conditions that UAS operates in such as adverse weather conditions, varying vehicle mishap characteristics and so on.

Simulation of normal operating conditions in the first case shows that the expected utility of package delivery to the original goal location is higher. By looking at the path characteristics, it can be inferred that although the shorter alternatives are present, the longer path avoiding the narrow valleys until the final approach is found to be better in this case. It is also important to note that constructed utility function implicitly states that the longer the path is, the higher chance to have a failure which decreases the final utility of the objective. Therefore, the decision here takes into account the duration of the path as well.

For the second case, increased failure rates pushes the planner to look for alternatives, which yielded the third location having the higher utility for package delivery among other locations. However, the utility found is still less than utility of ”Do-Not-Fly” action, shown in Fig 4(d). This means that flying to deliver a package in a given operating condition is actually less beneficial than staying on the ground. Hence, the decision for the mission should be do-not-attempt for delivery in this case.

It is noticed that the selection of utilities for the mission objectives to construct a desired scenario is not a trivial task and it also plays a crucial role on the path search and the task-level decisions. For the construction of complex and extensive mission scenarios, more detailed studies are required to establish the relations between utility values and the real-world mission objectives.

V. CONCLUSIONS

In this paper, we have proposed a generic multi-objective utility function for the path planning of UAS operations that can address the diverse mission objectives such as safety risks, property damage and the benefits of the mission so that the task-level decision can be made during an autonomous mission by the optimization of the utility function. Proposed concept is applied in a simple UAS package delivery scenario using multi-T-RRT* algorithm, and the results are discussed.

ACKNOWLEDGMENT

This study is supported by National Science Foundation (S&AS) under the grant 1724248.

REFERENCES

[1] E. Ancel, F. M. Capristan, J. V. Foster, and R. C. Condotta, "Real-time risk assessment framework for unmanned aircraft system (uas) traffic management (utm)," 06 2017.

[2] J. Rubio-Hervas, A. Gupta, and Y.-S. Ong, "Data-driven risk assessment and multicriteria optimization of uav operations," *Aerospace Science and Technology*, vol. 77, pp. 510 – 523, 2018.

[3] C. Wong, E. Yang, X.-T. Yan, and D. Gu, "Optimal path planning based on a multi-tree t-rrt* approach for robotic task planning in continuous cost spaces," 09 2018, pp. 242–247.

[4] T. Shan and B. Englot, "Sampling-based minimum risk path planning in multiobjective configuration spaces," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 814–821.

[5] U. C. Kaya, A. Dogan, and M. Huber, "A probabilistic risk assessment framework for the path planning of safe task-aware UAS operations," in *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, Jan 2019.

[6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[7] D. Devaurs, T. Siméon, and J. Cortés, "A multi-tree extension of the transition-based rrt: Application to ordering-and-pathfinding problems in continuous cost spaces," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2991–2996.