

Robot Risk-Awareness by Formal Risk Reasoning and Planning

Xuesu Xiao , Jan Dufek , and Robin R. Murphy

Abstract—This letter proposes a formal robot motion risk reasoning framework and develops a risk-aware path planner that minimizes the proposed risk. While robots locomoting in unstructured or confined environments face a variety of risk, existing risk only focuses on collision with obstacles. Such risk is currently only addressed in ad hoc manners. Without a formal definition, ill-supported properties, e.g. additive or Markovian, are simply assumed. Relied on an incomplete and inaccurate representation of risk, risk-aware planners use ad hoc risk functions or chance constraints to minimize risk. The former inevitably has low fidelity when modeling risk, while the latter conservatively generates feasible path within a probability bound. Using propositional logic and probability theory, the proposed motion risk reasoning framework is formal. Building upon a universe of risk elements of interest, three major risk categories, i.e. locale-, action-, and traverse-dependent, are introduced. A risk-aware planner is also developed to plan minimum risk path based on the newly proposed risk framework. Results of the risk reasoning and planning are validated in physical experiments in a real-world unstructured or confined environment. With the proposed fundamental risk reasoning framework, safety of robot locomotion could be explicitly reasoned, quantified, and compared. The risk-aware planner finds safe path in terms of the newly proposed risk framework and enables more risk-aware robot behavior in unstructured or confined environments.

Index Terms—Robotics in hazardous fields, robot safety, search and rescue robots.

I. INTRODUCTION

ROBOTS that locomote in unstructured or confined environments usually face motion risk. Therefore, autonomous planning systems must be capable of deciding how to reduce risk. This intelligent behavior firstly requires a fundamental understanding of risk and then a planning paradigm to find motion plans with minimum risk. The usages of unmanned vehicles in situations such as Urban Search And Rescue (USAR), nuclear operations, disaster robotics [1], etc., are examples where the execution of motion inherently entails taking risk and therefore motivate a formal risk reasoning framework and corresponding risk-aware planner.

Manuscript received September 8, 2019; accepted January 29, 2020. Date of publication February 17, 2020; date of current version February 26, 2020. This letter was recommended for publication by Associate Editor B. Morris and Editor Y. Choi upon evaluation of the reviewers' comments. This work was supported in part by NSF under Grant 1637955, NRI: A Collaborative Visual Assistant for Robot Operations in Unstructured or Confined Environments. (*Corresponding author: Xuesu Xiao.*)

Xuesu Xiao is with the Department of Computer Science, University of Texas at Austin, Austin, TX 78712 USA (e-mail: xiao@cs.utexas.edu).

Jan Dufek and Robin R. Murphy are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: dufek@tamu.edu; robin.r.murphy@tamu.edu).

Digital Object Identifier 10.1109/LRA.2020.2974434

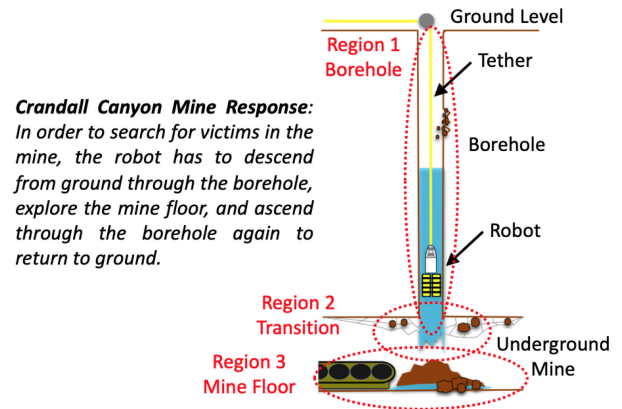


Fig. 1. Mine disaster borehole entry [2].

Fig. 1 shows an example of borehole entry from 2007 Crandall Canyon Mine response [2], where the robot was tasked to explore the underground mine floor to search for victims, and its entry and exit point was through a borehole from the surface. In region 1, the borehole area, small clearance of the borehole may cause the robot getting jammed. Due to the lack of casing of the borehole, falling rocks, drilling foam, water, and debris (from drilling the borehole) may damage the robot. The vertically hanging robot might spin and therefore lose controllability and mobility. In region 2, the transition from the borehole to the mine, the robot may get stuck with the mesh roof (existing infrastructure to keep the integrity of the mine) during hole exit and reentry. Tether entanglement with mesh roof is another risk. Furthermore, the transition from vertical mobility to operating on mine floor also requires extra effort and induces risk. Region 3, the inside of the mine, has unstable terrain due to running water and mud, causing the robot getting trapped. The robot also has to traverse soft drill tailings and foam (from drilling and mining activities), or even equipment, before reaching the mine floor. Also, while locomoting in region 3, robot tether is still being extended or retracted, interacting with the borehole (region 1) and the mesh roof in the transition into the mine (region 2). Risk of tether entanglement still exists. Due to the variety of existing risk sources in borehole entry, the robot failed at Crandall Canyon Mine in all four runs during the response in 2007 (lowering system failure, blockage in borehole, deteriorated sensing from water, debris, and drilling foam, and entangled and finally broken tether [2]).

Motivated by the question of how the variety of risk sources contribute to the high failure rate, this letter formally defines robot motion risk using propositional logic and probability theory. In contrast to the conventionally assumed additive or

Markovian properties, the formal methods used in the derivation reveals contradicting facts: risk's non-additivity and history dependency. Building upon a comprehensive universe of risk elements (not only obstacles), a variety of adverse effects which exist in unstructured or confined environments are categorized into locale-dependent, action-dependent, and traverse-dependent risk elements. The proposed risk-aware planner is able to take partial history information into account, and plans minimum risk path based on the newly proposed risk reasoning framework (for locale- and action-dependent risk elements). It also provides a paradigm to address risk with deeper history dependency (up to traverse). The risk reasoning and risk-aware planning results are validated through a physical robot flying in a real-world unstructured or confined environment.

The rest of the letter is organized as follows: Sec. II provides related work. Sec. III formally derives the proposed risk reasoning framework. Sec. IV presents the risk-aware planner and points out the tradeoff between computation complexity and history dependency depth. Sec. V presents an integrated demonstration of the risk reasoning and risk-aware planning results on a physical tethered Unmanned Aerial Vehicle (UAV) in a real-world unstructured or confined environment. Sec. VI concludes the letter.

II. RELATED WORK

This section reviews current approaches to reason about risk and existing risk-aware planners.

A. Risk Reasoning

Although risk has been addressed in the literature, a formal definition of risk is still unclear. Risk was referred to as some numerical measure of the severity/negativity related with certain aspects of motion. Risk associated with locomoting in unstructured or confined environments was explicitly represented as ad hoc risk functions of robot state.

[3] represented risk in the workspace by two layers: hazard data and visibility layer. The risk related with each layer was a function of the particular state. Weighted sum was used to combine the effect from both risk layers. To the author's best knowledge, this is the only work that considered more than one risk sources. [4] associated UAV flight risk at a certain location with this location's ground orography. [5] adopted a similar approach and also assumed risk to be a function of location only. Even with data-driven approaches, researchers estimated potential risk of a certain state based on historical record, including ocean Automated Identification System (AIS) [6] and traffic data [7].

Besides the lack of formal definition, all the risk considered in the literature was related with obstacles of various form. However, moving robots are also exposed to other kinds of risk, such as risk from turning, terrain interaction, battery condition, etc. Furthermore, the assumption that the risk a robot faces at a certain state/location is only a function of that state/location is ill-founded. So is the additivity of risk. Existing work assumed the risk a robot faces when executing a path is simply addition of the risk from all states.

B. Risk-Aware Planning

In search of a feasible path plan, risk-awareness was mostly modeled as (chance) constraints. [8] proposed

chance-constrained rapidly-exploring random tree (CC-RRT) approach, which used chance constraints to guarantee probabilistic feasibility at each time step and over entire trajectory. Another framework is (Partially Observable) Markov Decision Process ((PO)MDP). As standard MDP inherently contains reward but not risk, researchers have looked into representing risk as negative reward (penalty) [6] or constraints (C-POMDP) with unit cost for constraint violation [9]. Going beyond unit cost, CC-POMDP was proposed by [10], which was based on a bound on the probability (chance) of some event happening during policy execution. Robust Model Predictive Control (RMPC) is another alternative, with an emphasis on risk allocation, i.e., to allocate more risk for more rewarding actions [11], [12].

Existing constraint violation which models risk took the form of intersection between path and obstacles. A feasible plan was a path with constraint violation probability bounded by a certain manually assigned threshold, not with minimum risk in an absolute sense. Furthermore, the temporal or spatial (multiple obstacles) dependencies of constraint violation probability were either assumed to be independent or relaxed using ellipsoidal relaxation technique or Boole's inequality [8], [10]–[12]. This neglected the important dependencies on the motion history and the rough approximation introduced significant conservatism.

III. RISK REASONING FRAMEWORK

This section formally defines robot motion risk in unstructured or confined environments and derives the reasoning framework using propositional logic and probability theory. It is comprehensive and general for any robots.

A. Formal Definition and Explicit Representation

This work considers motion risk for mobile robots executing a path. Risk in terms of a sequence of motion (path) is formally defined as *the probability of the robot not being able to finish the path*.

The robot workspace is defined based on tessellation of the Cartesian space. Each tessellation is either a viable (e.g. free) or unviable (e.g. occupied) state for the robot to locomote. A feasible path plan P is defined to be an ordered sequence of viable tessellations, called states and denoted as s_i :

$$P = \{s_0, s_1, \dots, s_n\}, \|s_i - s_{i-1}\|_2 \leq r_c, \forall 1 \leq i \leq n$$

where r_c is the maximum distance between two consecutive states for the path to be feasible. A state on the path is finished by the robot reaching it within an acceptable tolerance and ready to move on to the next state. A state is not finished due to two main reasons: the robot crashes or gets stuck. In order to finish the path of n states, the robot faces r different risk elements, which will possibly cause not finishing the path. Here, three types of events are defined with propositional logic:

- F – the event where the robot finishes path P
- F_i – the event where the robot finishes state i
- F_i^k – the event where risk k does not cause a failure at state i

The reasoning about motion risk is based on three assumptions, which are expressed by propositional logic:

- 1) Path is finished only when all states are finished:

$$F = F_n \cap F_{n-1} \cap \dots \cap F_1 \cap F_0$$

- 2) A state is finished only when all risk elements do not cause failure:

$$F_i = F_i^1 \cap F_i^2 \cap \dots \cap F_i^{r-1} \cap F_i^r$$

- 3) Finish or fail a state because of one risk element is conditionally independent of finish or fail that state because of any other risk element, given the history leading to the state:

$$\left(F_i^1 \mid \bigcap_{j=0}^{i-1} F_j \right) \perp\!\!\!\perp \left(F_i^2 \mid \bigcap_{j=0}^{i-1} F_j \right) \perp\!\!\!\perp \dots \perp\!\!\!\perp \left(F_i^r \mid \bigcap_{j=0}^{i-1} F_j \right)$$

where $\perp\!\!\!\perp$ means “independent”.¹ Before reasoning about risk, the probability of the robot *being* able to finish the path $P(F)$ is firstly reasoned:

$$\begin{aligned} P(F) &= P(F_n \cap F_{n-1} \cap \dots \cap F_0) \\ &= P(F_n | F_{n-1} \cap \dots \cap F_0) \cdot \dots \cdot P(F_1 | F_0) \cdot P(F_0) \\ &= \prod_{i=0}^n P \left(F_i \mid \bigcap_{j=0}^{i-1} F_j \right) \\ &= \prod_{i=0}^n P \left(F_i^1 \cap F_i^2 \cap \dots \cap F_i^r \mid \bigcap_{j=0}^{i-1} F_j \right) \\ &= \prod_{i=0}^n P \left(F_i^1 \mid \bigcap_{j=0}^{i-1} F_j \right) \cdot \dots \cdot P \left(F_i^r \mid \bigcap_{j=0}^{i-1} F_j \right) \\ &= \prod_{i=0}^n \prod_{k=1}^r P \left(F_i^k \mid \bigcap_{j=0}^{i-1} F_j \right) \end{aligned} \quad (1)$$

The first, second, fourth, and fifth equal signs are due to assumption 1, probability chain rule, assumption 2, and assumption 3, respectively. Therefore, the formal risk definition, the probability of *not* being able to finish the path, is the probabilistic complement:

$$\begin{aligned} P(\bar{F}) &= 1 - P(F) \\ &= 1 - \prod_{i=0}^n \prod_{k=1}^r P \left(F_i^k \mid \bigcap_{j=0}^{i-1} F_j \right) \\ &= 1 - \prod_{i=0}^n \prod_{k=1}^r \left(1 - P \left(\bar{F}_i^k \mid \bigcap_{j=0}^{i-1} F_j \right) \right) \end{aligned} \quad (2)$$

In terms of risk representation, the risk of path P is denoted as $risk(P)$ and is equal to $P(\bar{F})$. $P(\bar{F}_i^k | \bigcap_{j=0}^{i-1} F_j)$ means the probability of risk k causes failure at state i , given the history of finishing s_0 to s_{i-1} . It is therefore denoted as the k th risk robot faces at state i given that s_0 to s_{i-1} were finished: $r_k(\{s_0, s_1, \dots, s_i\})$. Writing in risk representation form will

¹This simplified assumption may not always hold when more realistic and complicated risk inter-dependency models are necessary.

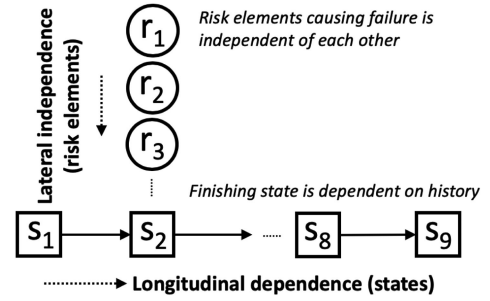


Fig. 2. Longitudinal Dependence on History States and Lateral Independence among Risk Elements.

Universe of all risk elements

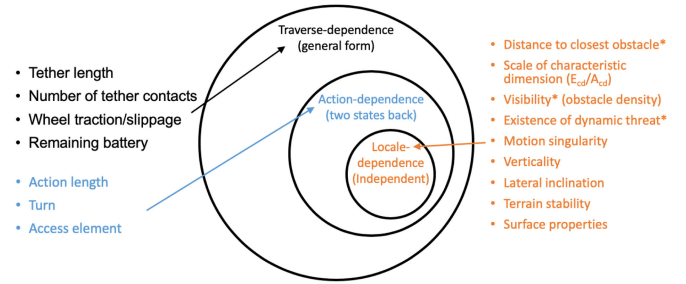


Fig. 3. Universe of all risk elements: Three categories with examples. Subset relationships are shown in the Venn diagram.

yield:

$$risk(P) = 1 - \prod_{i=0}^n \prod_{k=1}^r (1 - r_k(\{s_0, s_1, \dots, s_i\})) \quad (3)$$

This proposed probabilistic motion risk framework does not require the ill-founded additivity assumption for risk. More importantly, the conditional probability in Eqn. (2) clearly shows the dependency of risk at certain state on the history, not only the state itself. Despite the dependencies in the temporal domain, conditional independence among different risk elements at a certain state given the history is still assumed. Along the direction of the path, risk the robot faces at each individual state is dependent on history (longitudinal dependence), while at each state, risk caused by different risk elements is independent (lateral independence) (Fig. 2). Note that each individual risk values $r_k(\{s_0, s_1, \dots, s_i\})$ could be determined theoretically or empirically.

B. Risk Categories and Risk Elements

The formal definition and explicit representation reveal the longitudinal dependence of risk at a certain state on the history. Mathematically speaking, the dependency is on the entire history in general. However, in practice, the dependency of different risk elements may have different depth into the history, e.g. crash to a close obstacle is only dependent on the closeness of this state to obstacle, or crash due to an aggressive turn (change of action direction) is only dependent on the current and last action taken. In this work, risk elements are divided into three categories: *locale*-dependent, *action*-dependent, and *traverse*-dependent risk elements. Fig. 3 shows the universe of all risk

elements with 16 examples, and the categories they belong to. More importantly, the subset relationship between the three categories are displayed: locale-dependence \subset action-dependence \subset traverse-dependence. This universe is not exclusive.

1) *Locale-Dependence*: Risk elements that do not depend on history at all. This most special case of history dependence allows relaxation of the entire history dependency:

$$P\left(\bar{F}_i^k \mid \bigcap_{j=0}^{i-1} F_j\right) = P(\bar{F}_i^k) \quad (4)$$

The word *locale* connotes the meaning of “location,” “position,” or where the robot is currently at. It has similar connotation as the concept of “state” in (Cartesian) configuration space, but also emphasizes the relationship with the current proximity of the environment. This category of risk elements has been covered in existing literature under the name of “location” or “state” and was assumed to be the only type of risk elements. This type of traditional risk elements could be evaluated on the state alone, not depending on history.

2) *Action-Dependence*: Risk elements that depend on two previous states. This special case of risk’s history dependency, between the general traverse-dependence and the most special locale-dependence, traces two states back into the history, such that the finishing of the last two states has impact on the risk the robot is facing at the current state:

$$P\left(\bar{F}_i^k \mid \bigcap_{j=0}^{i-1} F_j\right) = P(\bar{F}_i^k | F_{i-2} \cap F_{i-1}) \quad (5)$$

This category of risk elements usually focuses on the transitions between states, including the effort necessary to initiate the transition and the difference between two consecutive transitions.

3) *Traverse-Dependence*: Risk elements that depend on the entire history. This is the general form of risk’s history dependency, which encompasses both locale-dependent and action-dependent risk elements:

$$P\left(\bar{F}_i^k \mid \bigcap_{j=0}^{i-1} F_j\right) = P(\bar{F}_i^k | F_{i-1} \cap F_{i-2} \cap \dots \cap F_1 \cap F_0) \quad (6)$$

The general form has a full depth of history dependency and looks back to the whole traverse from start leading to the current state. Finishing of all the history states has impact on the finishing of the current state.

IV. RISK-AWARE PLANNING

This section develops a risk-aware planner that plans with the newly proposed risk reasoning framework. It guarantees optimality for locale- and action-dependent risk elements. Tradeoff between computation complexity and history dependency depth is discussed.

A. Impact of Non-Additivity and History-Dependency

The risk-aware planner needs to find minimum risk path between a start and a goal location. Traditional risk-aware planners assumed additive and Markovian properties, therefore their cost

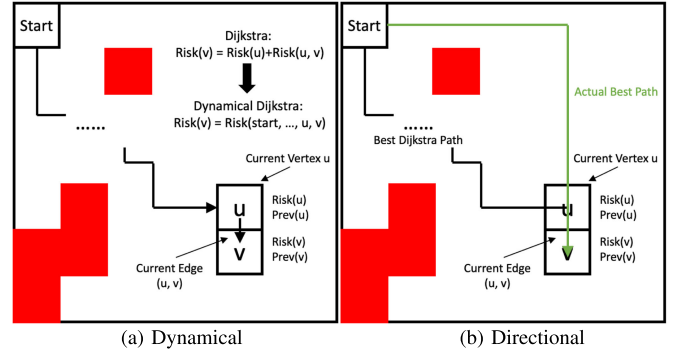


Fig. 4. Dynamical and directional risk evaluation: Path risk needs to be evaluated dynamically and non-optimal substructure requires minimum risk path to be directional.

function is simply:

$$risk(P) = \sum_{i=0}^{i=1} r(s_i) \quad (7)$$

Nonetheless, the cost function based on the proposed risk framework is Eqn. (3).

This risk representation has neither additivity nor state-dependency, and therefore does not have substructure optimality. The risk robot faces at state i is not well-defined on s_i , but can take different values depending on the traverse taken $\{s_0, s_1, \dots, s_{i-1}, s_i\}$.

In terms of the impact of those differences on the planner, an intuitive visual example is shown in Fig. 4(a): when traditional approaches (such as Dijkstra’s algorithm) expand from vertex u to vertex v , the risk of the path from start to v is simply the sum of the risk of the subpath from start to u and the risk at v . The risk at v is simply well defined on vertex v . However, using the proposed risk representation (Eqn. (3)), the risk of vertex (state) v is not well defined by only looking at v alone. The dependency on the history requires the risk at v to be evaluated based on the entire traverse (start, ..., u, v).

Due to the dynamically changing state risk dependent on history, the problem loses optimal substructure, therefore scenarios such as the one shown in Fig. 4(b) may occur: traditional approaches base on substructure optimality, so if the optimal path to v passes through u , the subpath to u is also guaranteed to be optimal to u . However, due to the risk representation in Eqn. (3), this may not be necessarily true. Given the green path is the best path to v , the subpath of the green path to u is not the best path to u . The black path actually is. The optimal path to v is from a different direction to u as the optimal path to u .

Therefore the planning problem at hand has at least two issues, preventing from the usage of traditional search algorithms: the risk being dynamical (dependent on history) and directional (lost optimal substructure). Considering the three categories of risk elements, locale-dependency, action-dependency, and traverse-dependency, a new algorithm is designed to optimally address the first two categories of risk elements, by looking back two states into history during planning, at the cost of more computation. Traverse-dependent risk elements, however, cannot be guaranteed to be optimally addressed, since it is the most general form of risk and the look-back has to be into the

Algorithm 1: Risk-Aware Path Planner.

Input: \mathcal{G}, v_{start}
Output: Risk-Aware paths to all vertices other than v_{start}

- 1: $\forall D_i^{(j)} \in \mathcal{D}$ set $r_i^{(j)} \leftarrow \infty$ and $PD_i^{(j)} \leftarrow NULL$
- 2: For v_{start} , set $r_{start}^{(j)} \leftarrow 0$ in all $D_{start}^{(j)}$
- 3: Initialize visited set to $\mathcal{R} \leftarrow \{\}$
- 4: **while** $\mathcal{R} \neq \mathcal{D}$ **do**
- 5: pick vertex v_u with smallest $r_u^{(i)}$ where $D_u^{(i)} \notin \mathcal{R}$
- 6: **for** each edge $(v_u, v_v) \in \mathcal{E}$ **do**
- 7: $path_u^{(i)} \leftarrow backtrack(D_u^{(i)})$
- 8: $path_v(i) \leftarrow path_u^{(i)} \cup \{v_v\}$
- 9: $path_risk_v(i) \leftarrow evaluate(path_v(i))$
- 10: $current_min_risk \leftarrow v_v.D_v^{(j)}.r_v^{(j)}$, where $D_v^{(j)}$ corresponds to reaching v_v from v_u
- 11: **if** $path_risk_v(i) < current_min_risk$ **then**
- 12: $v_v.D_v^{(j)}.r_v^{(j)} \leftarrow path_risk_v(i)$
- 13: $v_v.D_v^{(j)}.PD_v^{(j)} \leftarrow D_u^{(i)}$
- 14: **end if**
- 15: **end for**
- 16: $\mathcal{R} \leftarrow \mathcal{R} \cup \{D_u^{(i)}\}$
- 17: **end while**
- 18: **for** each $v_i \in \mathcal{V}$ **do**
- 19: pick $D_i^{(j)}$ with the smallest $r_i^{(j)}$
- 20: $risk_i \leftarrow r_i^{(j)}$
- 21: $path_i \leftarrow backtrack(D_i^{(j)})$
- 22: **end for**
- 23: **return** all $path_i$ with $risk_i$

entire history to guarantee optimality. Only those risk caused by two steps back into the traverse could be handled.

B. Risk-Aware Planner

The tessellated workspace is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ be the vertex set, and $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ be all the edges connecting the vertices. To accommodate the history dependency of action-dependent risk elements, each vertex is further represented by $v_i = (D_i^{(1)}, D_i^{(2)}, \dots, D_i^{(c)})$, where $D_i^{(j)}$ represents the direction from which v_i is reached. They memorize the two-step history information to be used when being expanded in the future. The total number c is the connectivity of v_i , as the number of incoming edges reaching v_i . For each direction reaching v_i , $D_i^{(j)}$ is defined as $D_i^{(j)} = (r_i^{(j)}, PD_i^{(j)})$, where $r_i^{(j)}$ is the risk of reaching v_i from direction $D_i^{(j)}$ starting from start vertex v_{start} , and $PD_i^{(j)}$ is the previous direction of reaching the previous vertex, in other words, previous direction of two steps back. All the directions of all vertices $D_i^{(j)}$ compose the superset of all directions $\mathcal{D} = \{D_i^{(j)} | i = 1, 2, \dots, n\}$ and j is a variable for different vertices depending on how many directions (edges) are leading to the vertex. The algorithm is shown in Alg. 1. It finds the minimum-risk directional component in the graph (line 5) and expands the vertex which this directional component belongs to (line 6 - line 15). After expanding all the neighbors, this directional component is marked visited (line 16). When all directional components are visited, the final minimum-risk

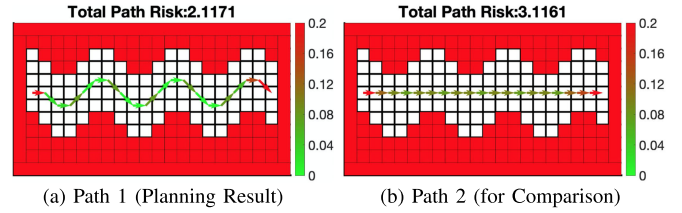


Fig. 5. Conventional planner with additive state-dependent risk.

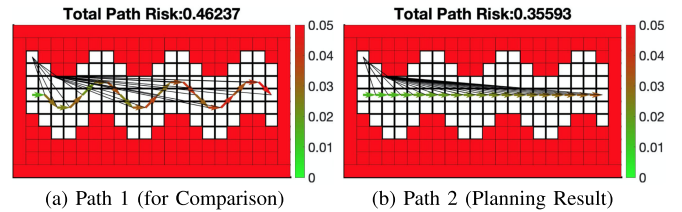


Fig. 6. Proposed planner with probabilistic history-dependent risk.

path to each vertex is selected from its minimum-risk directional components (line 18–22).

C. Risk Representation and Planning Examples

Taking a tethered robot locomoting in obstacle-occupied (shown as red) 2D space as example, results from the proposed risk representation and risk-aware planner are shown in Fig. 6. As comparison, results of conventional risk-aware planner based on additive state-dependent risk are presented in Fig. 5. The color of the arrows indicates the conventional and proposed risk the robot faces at each state and the color map is displayed on the right. The robot starts from the left and the goal is going to the right. Six risk elements are chosen as examples from the three risk categories: distance to closest obstacle and visibility from locale-dependent risk elements, action length and turn from action-dependent risk elements, and tether length and number of tether contacts [13] from traverse-dependent risk elements. Risk caused by each risk element at each state is determined empirically, e.g. a state which is closer to obstacle, requires a sharper turn, or involves more contact points is assigned a higher risk value.

Fig. 5(a) shows the result of conventional risk-aware planner using additive state-dependent risk. Due to the assumption of state-dependency, action length, turn, and tether length, number of contacts cannot be properly addressed by the planner. The only possible risk elements are distance to closest obstacle and visibility, which are evaluated based on state alone. Their risk values at each state are combined using normalization and weighted sum (identical weights for both risk elements in the examples) and summed up along the entire path. Using conventional approach, the planner will find the path shown in Fig. 5(a), since this is the minimum risk path according to the conventional additive state-dependent risk representation and could be found by traditional search-based algorithms, such as Dijkstra's or A*. The path shown in Fig. 5(b), however, will be neglected, since it is supposed to have a higher risk according to the additive state-dependent risk representation.

Fig. 6(b) shows the result of the proposed risk-aware planner using probabilistic history-dependent risk, which reveals

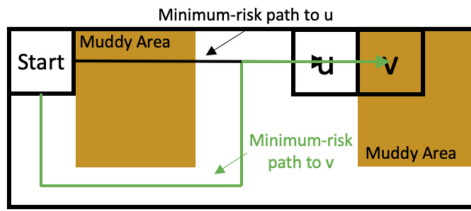


Fig. 7. Failure Case due to Traverse-Dependency.

the actual higher risk in Fig. 6(a). All six risk elements from all three risk categories could be properly addressed by the proposed planner, with the optimality of locale-dependent and action-dependent risk elements guaranteed. The risk at each state is now formulated as the probability of the robot not being able to finish the state, displayed in color. The probability of not being able to finish the path, as risk of the path, is computed using Eqn. (3). The two-step look-back in the proposed risk-aware planner makes sure that history dependencies of risk up to actions could be addressed optimally. The traverse-dependent risk elements, however, are only suboptimal, or in other words, optimal up to two states in the history of the traverse, not the entire history to the start. As shown by Fig. 6, the risk aware-planner is willing to sacrifice distance to closest obstacle and visibility (locale-dependent) for shorter action length, less aggressive turn (action-dependent), and shorter tether length, fewer tether contacts (traverse-dependent). Note the computation complexity increases from $\mathcal{O}(V^2)$ (Fig. 5(a)) to $\mathcal{O}(CV^3)$ (number of vertices V , connectivity $C = 8$ in Fig. 6(b)), which will be discussed in detail in the next subsection.

One example of why the proposed risk-aware planner cannot optimally address traverse-dependent risk elements is shown in Fig. 7. Take wheel traction/slippage as an example of traverse-dependent risk element and assume the robot has two muddy areas to negotiate with in the workspace: the minimum risk path to u could be the black path, since u is in a clean area and the mud built up on the robot wheels would not cause significant risk at u . However, if the robot keeps venturing into v , which is another muddy area, the mud built up on the wheels from the first muddy area may cause major risk and the robot has very high probability of getting stuck at v . The green path becomes less risky, since the risk associated with the extra length and turns are justified by keeping clean wheels and reliable traction. However, the green path can never be found by the proposed risk-aware planner, since two-step look-back (from v looking back to u and the state left to u) cannot cover sufficient depth into history to find the green path. Therefore, for traverse-dependent risk element, only risk caused by the last two steps could be properly addressed, in the similar way as how action-dependent risk element is addressed.

D. Tradeoff: Computation Complexity vs. Dependency Depth

Although it is shown above that the proposed risk-aware planner is not optimal with respect to general traverse-dependent risk elements, it is possible that look-back into more steps in the history can direct the planner closer to the true optimal path, but at the cost of computation. Fig. 8 shows the potential extension of the proposed risk-aware planner in order to be able to address more depth in history dependency. The proposed risk-aware planner looks two-step back into the history and

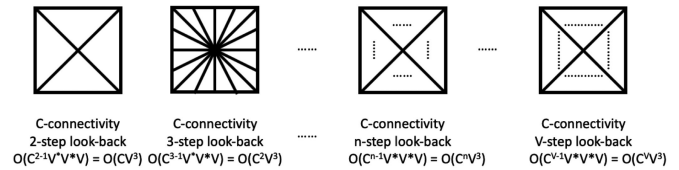


Fig. 8. Potential extension of the proposed risk-aware planner: Trading more computation for deeper history dependency. Graphical illustrations assume 4-connectivity as example.

therefore augments every original vertex into C directional components (as the four partitions in the left state of Fig. 8, assuming $C = 4$). If three-step look-back is necessary, the original vertex could be augmented into C^2 directional components (as the sixteen partitions in the second to left state of Fig. 8, assuming $C = 4$). By the same token, an arbitrary number n -step look-back requires C^{n-1} directional components. The deepest possible history dependency is V steps, as the longest simple paths have V vertices (here, V is trivially equivalent to $V - 1$) and the complexity would be $\mathcal{O}(C^V V^3)$. The complexities of the proposed risk-aware path planner (2-step look-back), potential 3-step look-back planner, general risk-aware planner (n -step look-back), and omnipotent risk-aware planner with full history dependency are shown in Fig. 8. The omnipotent risk-aware planner with full history dependency is supposed to guarantee optimality with even traverse-dependent risk elements, but the computation is intractable.

V. INTEGRATED DEMONSTRATION

This section presents an integrated demonstration using physical robot locomoting in a real-world unstructured or confined environment. Risk of two paths is reasoned and then compared. Physical execution of the paths finally validates the proposed risk-awareness.

A. Robot Mission

A tethered UAV working as an aerial visual assistant for a tele-operated primary ground robot [14] in a confined staircase is used to validate the results. The marsupial robot team locates at the second level of the staircase. The teleoperation task of the primary ground robot is to insert a sensor between stair railings and drop it into a pool of contamination at the bottom for radioactivity strength reading. This resembles a scenario in Fukushima Daiichi nuclear disaster response.

The map of the environment is given and there exist two good viewpoints to visually assist the sensor insertion task at the point of interest. Based on the proposed risk reasoning framework, the risk-aware planner needs to plan two paths from the UAV initial location to the two good visual assistance viewpoints (Fig. 9) and the less risky path is preferred. Only UAV motion is considered for the purpose of demonstration of risk-awareness.

B. Risk Reasoning and Risk-Aware Planning Results

Six different risk elements are considered: distance to closest obstacle (Dist.) and visibility (Vis.) (locale-dependent), action length (A. L.) and turn (action-dependent), tether length (T. L.) and number of tether contacts (Cont. #) (traverse-dependent). The choice of these six risk elements are due to their relevance

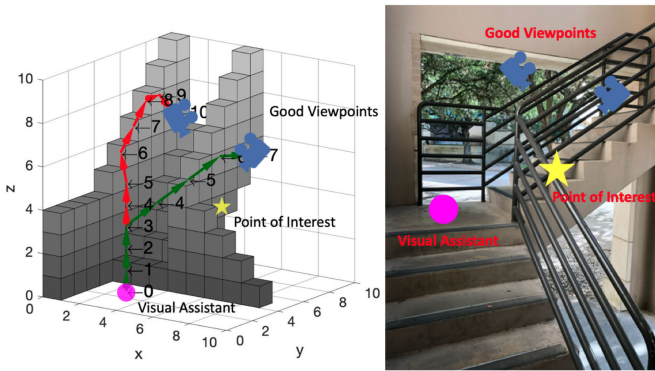


Fig. 9. Greyscale voxels represent occupied spaces. Magenta circle shows initial UAV location. Yellow star is the Point of Interest, i.e. the insertion point of the manipulator arm between the railings. The two cameras represent the two good viewpoints.

TABLE I
DETAILED RISK REPRESENTATION FOR RED PATH

Index	Dist.	Vis.	A. L.	Turn	T. L.	Cont. #	State
0	0.01	0.02	0	0	0.01	0	0.04
1	0.01	0.02	0.04	0	0.01	0	0.08
2	0.01	0.02	0.04	0	0.01	0	0.08
3	0.01	0.02	0.04	0	0.01	0	0.08
4	0.01	0.01	0.04	0	0.02	0	0.08
5	0.01	0.01	0.04	0	0.02	0	0.08
6	0.01	0.01	0.06	0.05	0.02	0	0.14
7	0.01	0.01	0.05	0.05	0.03	0	0.14
8	0.01	0.01	0.05	0	0.04	0	0.11
9	0.02	0.02	0.04	0.05	0.04	0	0.16
10	0.03	0.04	0.05	0.05	0.04	0	0.19

to this particular robot platform in this particular unstructured or confined environment, the practicality or availability of necessary risk information, and the representativeness of the three major risk categories. The minimum risk path to the two good viewpoints are planned, shown in Fig. 9 left.

The red path aims at one good viewpoint between the two staircase railings. Since a direct path needs to go through spaces confined by staircase railings and walls, the UAV needs to maneuver through those spaces to remain far away from obstacles and high visibility, but at the cost of a longer path and more turns. The risk associated with the red path is evaluated to be 0.714. The detailed risk representation for each state and each individual risk element on the red path is shown in Table I. The last column shows the state risk the robot faces at each state.

The green path aims at the other viewpoint in the wide open space in the middle of the staircase. Going there straight from the initial location needs to closely pass by the top of the railings. The planner chooses to make a slight detour to enlarge the clearance. However, maximizing distance and visibility has longer path and more turns as cost, so the planner chooses a compromise in between, shown as the 45° middle segment on the green path: the UAV does not fully sacrifice path length and twistiness for clearance, so it cuts through the free space with a straighter path and slightly (not completely) avoids the obstacles. The risk associated with the green path is evaluated to be 0.575 (computed from Table II). No contact points are formed in both cases.

TABLE II
DETAILED RISK REPRESENTATION FOR GREEN PATH

Index	Dist.	Vis.	A. L.	Turn	T. L.	Cont. #	State
0	0.01	0.02	0	0	0.01	0	0.04
1	0.01	0.02	0.04	0	0.01	0	0.08
2	0.01	0.02	0.04	0	0.01	0	0.08
3	0.01	0.02	0.04	0	0.01	0	0.08
4	0.03	0.01	0.06	0.05	0.02	0	0.16
5	0.04	0.01	0.06	0	0.02	0	0.12
6	0.02	0.01	0.06	0	0.03	0	0.12
7	0.01	0	0.05	0.05	0.03	0	0.13

TABLE III
EXPERIMENTAL TRIALS AND SUCCESS/FAILURE

Trial #	Red Path	Green Path
1	Collision w. railings	Collision w. wall
2	Oscillation	Success
3	Success	Oscillation
4	Collision w. wall	Success
5	Oscillation	Collision w. railings
6	Oscillation	Success
7	Success	Collision w. wall
8	Collision w. wall	Contacts formed, localization lost
9	Collision w. railings	Success
10	Oscillation	Contacts formed, localization lost
Fail. Rate	0.8	0.6

It is worth to note that using the traditional state-dependent only risk representation, the red path has a lower additive risk, because it maintains a relatively low state-dependent risk at most of the states on the path. The green path, however, would have higher risk, due to the compromise of locale-dependent (distance and visibility) for action-dependent risk elements (action length and turns). Although overall the compromise reduces path risk, it cannot be reflected by the traditional state-dependent risk representation.

C. Experimental Results and Discussions

Ten experimental trials each are conducted autonomously for the two paths. Path execution is manually terminated when the tethered UAV is unable to finish a state (collision or stuck). Table III shows the results of the 20 trials, either success or failure. The reasons of failure are specified.

For the red path, only two out of the ten experimental trials are successful. The other eight trials fail due to different reasons: trial 1, 4, 8, and 9 fail because of collision with railings or wall. Another reason for failure is oscillation. This happens primarily when the UAV is maneuvering to maintain a high obstacle clearance. The turning and long path have the potential of inducing extra turbulence in the confined staircase, therefore the rotorcraft can no longer maintain stability. Oscillation leads to collision or not being able to reach a certain state. The 80% failure rate is close to the 0.714 risk. Green path execution achieves 60% failure rate, which is also close to the 0.575 risk value. While in trial 1, 5, and 7 the UAV collides with the obstacles and it starts oscillate in trial 3, another important failure reason is the accidentally formed tether contact due to the closeness to the railings. But overall speaking, the relatively open space in the center of the staircase and the straightness and shortness contribute to a less risky path. Although the risk value caused by each individual risk element is only an empirical estimation, six and eight failures out of ten trials are sufficiently close

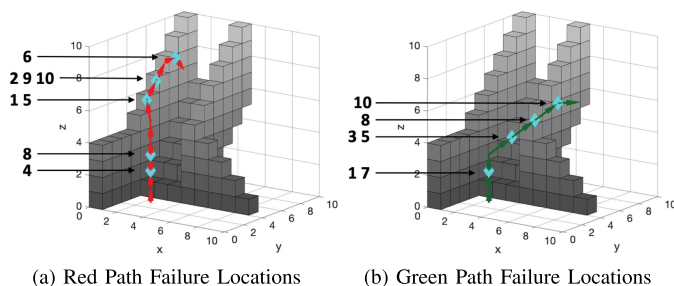


Fig. 10. Failure Locations on Both Paths: The numbers correspond to the trial number in Table III, indicating this particular trial is terminated at the state denoted by cyan diamonds.

to the 0.575 and 0.714 risk value, respectively. The proposed theoretical risk reasoning framework matches closely with the real-world failure rate.

Fig. 10 shows the locations of failure. The numbers on the left correspond to the failure trial numbers in Table III. Some failure locations only have one failure trial, while others may have multiple. Most failure locations for both cases are in the top part of the path, due to either complex trajectory shape (longer path and more turns for red path) or closeness to obstacles (collision or tether contact with obstacles for green path). It roughly matches with the state risk values in the last column of Table I and Table II: the high state risk values are correlated with more failure cases at that particular state in the physical experiments. Inspecting the failure reasons (Table III) and failure locations (Fig. 10), it could be seen that for red path most failures are caused by action-dependent risk elements while the effect of locale-dependent risk elements is minimized. But for green path, due to the sacrifice of locale-dependent risk elements for shorter path length and fewer turns (action-dependent risk elements), obstacles near states cause more possibility of failure to finish the path.

VI. CONCLUSION

This letter proposes a robot motion risk reasoning framework using propositional logic and probability theory. Risk is formally defined as *the probability of the robot not being able to finish the path*. The use of formal methods reveals risk's non-additivity and history-dependency, which was assumed otherwise in the existing literature. Built upon a comprehensive risk universe, three categories of risk elements are introduced: locale-, action-, and traverse-dependent. Based on the proposed risk reasoning

framework, a risk-aware planner is developed to take risk's newly discovered properties into account. It finds minimum risk path for locale- and action-dependent risk elements. But for traverse-dependency, optimality cannot be guaranteed. The trade-off between computation complexity and history dependency depth is discussed. The results of applying the risk reasoning framework and risk-aware planner on a physical tethered UAV flying in a real-world unstructured or confined environment are compared with the proposed theory, and the actual motion failure rate could be roughly reflected by the risk value. Future work will focus on developing better models to compute risk values for individual risk elements and more efficient approaches to extend history dependency depth.

REFERENCES

- [1] R. R. Murphy, *Disaster Robotics*. Cambridge, MA, USA: MIT Press, 2014.
- [2] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 91–103, Jun. 2009.
- [3] A. Soltani and T. Fernando, "A fuzzy based multi-objective path planning of construction sites," *Autom. Construction*, vol. 13, no. 6, pp. 717–734, 2004.
- [4] L. De Filippis, G. Guglieri, and F. Quagliotti, "A minimum risk approach for path planning of uavs," *J. Intell. Robot. Syst.*, vol. 61, no. 1-4, pp. 203–219, 2011.
- [5] M. Zabarankin, S. Uryasev, and P. Pardalos, "Optimal risk path algorithms," in *Cooperative Control and Optimization*. Berlin, Germany: Springer, 2002, pp. 273–298.
- [6] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *J. Field Robot.*, vol. 30, no. 5, pp. 741–762, 2013.
- [7] J. Krumm and E. Horvitz, "Risk-aware planning: Methods and case study for safer driving routes," in *Proc. AAAI*, 2017, pp. 4708–4714.
- [8] B. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2010, Art. no. 8160.
- [9] A. Undurti and J. P. How, "An online algorithm for constrained POMDPs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3966–3973.
- [10] P. Santana, S. Thiébaux, and B. Williams, "Rao*: An algorithm for chance constrained POMDPs," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3308–3314.
- [11] M. Ono and B. C. Williams, "An efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure," in *Proc. AAAI*, 2008, pp. 1376–1382.
- [12] M. P. Vitus and C. J. Tomlin, "On feedback design and risk allocation in chance constrained control," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 734–739.
- [13] X. Xiao, J. Dufek, M. Suhail, and R. Murphy, "Motion planning for a UAV with a straight or kinked tether," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 8486–8492.
- [14] X. Xiao, J. Dufek, and R. R. Murphy, "Autonomous visual assistance for robot operations using a tethered UAV," in *Field and Service Robotics*. Berlin, Germany: Springer, 2019.