2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Heterogeneous System of Systems Framework for Proactive Path Planning of a UAV-assisted UGV in Uncertain Environments

Patrick Sherman and Nicola Bezzo

***Abstract*— A common challenge for mobile robots is traversing uncertain environments containing obstacles, rough terrain, or hazards. Without full knowledge of the environment, an unmanned ground vehicle (UGV) navigating towards a goal could easily drive down a path that is blocked (requiring the robot to retrace sections of its path) or run into a hazard causing a catastrophic failure. To address this issue we propose a *system of systems* (SoS) abstraction to group a distributed set of robots into a single system. Specifically, we propose augmenting the sensing capabilities of a UGV using an unmanned aerial vehicle (UAV). With different dynamic and sensing capabilities, the UAV scouts ahead and proactively updates the plan for the UGV using information discovered about the environment. To predict reachable states of the UGV, the UAV employs a sampling-based method in which a set of virtual particles representing simulated instances of the UGV are used to approximate the distribution of possible trajectories. The UAV assesses if the current UGV path plan is inefficient or unsafe, and if so, provides an alternative path to the UGV. For robustness, a model predictive path integral (MPPI) optimization method is used to modify the waypoints when delivered to the UGV. The strategy is validated in simulation and experimentally.**

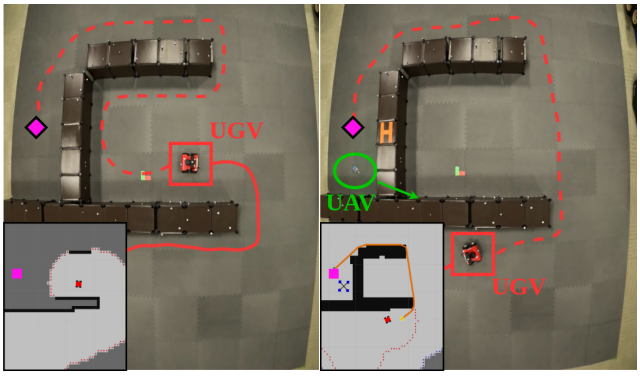
***Note*—Videos of the simulations and experiments are provided in the supplementary material and can be accessed at: `https://www.bezzorobotics.com/ps-iros24`.**

I. Introduction

The ability to develop autonomous solutions using a team of heterogeneous robots has the potential to solve a variety of real-world applications. Intuitively, a system of heterogeneous robots can take advantage of the unique capabilities of each robot type to complete tasks more efficiently than a single robot or a homogeneous multi-robot system. In this work, we focus on a particular heterogeneous system consisting of an unmanned ground vehicle (UGV) and an unmanned aerial vehicle (UAV) offering different but complementary motion and sensing capabilities.

Let us consider a UGV navigating alone to a goal position through an uncertain environment. Object detection and obstacle avoidance is achieved by using sensors like LiDARs or cameras. However, since such sensors are statically attached to the robot, it must rely on its mobility to detect and map the surrounding environment. With this constraint, the robot cannot effectively plan beyond its field of view, potentially causing drastic inefficiencies. For example, Fig. 1(a) shows a UGV navigating towards a goal. Currently unknown to the UGV, an obstacle blocks the path and the robot will be forced to turn around to find an alternate route. In other scenarios, there is also the possibility of damaging the UGV if it drives into a hazard that is undetectable by the sensors, such as a deep body of water or a cliff.

Patrick Sherman and Nicola Bezzo are with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA. Email: {ukw4tc, nb6be}@virginia.edu



(a) (b)

Fig. 1. UGV navigating in uncertain environment: (a) without assistance from UAV and (b) rerouted to shorter path after UAV has scouted ahead.

To address this problem, we propose a novel *system of systems* (SoS) architecture for proactive path planning that considers both liveness (i.e. something good will eventually happen) and safety (i.e. nothing bad will ever happen) constraints. This framework is a synthesis of proven robotic planning and sampling-based techniques that we combine into a high-level strategy for collaboration between a UAV and a UGV. By treating the UAV as an extension of the UGV's sensing and computation capabilities, the two distributed systems create a unified and more capable robotic system. Consider the result in Fig. 1(b). By scouting ahead to explore the environment, the UAV is able to discover and reroute the UGV to a shorter total path than the equivalent case in which the UGV was operating alone (Fig. 1(a)). To explore most effectively, the UAV scouts ahead along the predicted trajectory of the UGV. For trajectory prediction we propose a sampling-based reachability analysis technique that leverages virtual particles representing possible future beliefs of the UGV. When the UAV determines a better path for the UGV is possible, it can proactively reroute the UGV during operation. Dealing with the constraint that the UAV and UGV can be disconnected while exploring, the UAV is triggered to rendezvous with and reroute the UGV when: 1) a self-triggered signal if a more efficient path is discovered or 2) an event-triggered signal if the UGV is about to run into a hazard or is reaching a location where there are uncertainties on the direction it may take (e.g. a fork in the path). Once triggered, the UAV flies back to the UGV to reroute the ground robot to the better path by delivering an updated set of target waypoints. Finally, for additional robustness the UAV uses a model predictive path integral (MPPI) sampling based optimization technique to adjust the waypoints.

The main contribution of this work is 1) a coordinated system of systems (SoS) abstraction for autonomous path planning and decision making for safe and efficient navigation of a UGV dynamically rerouted by a UAV. Part of our

contribution also includes 2) a sampling based reachability analysis technique to predict the future closed-loop robot trajectories under model and noise uncertainties. In this paper, we specify a UGV-UAV system, however the proposed framework is general for different heterogeneous systems.

The rest of this paper is organized as follows. We first review related work in Sec. II. In Sec. III we mention our assumptions and constraints before defining the problem in Sec. IV. Next, in Sec. V we detail the complete approach of our framework. Finally, we present simulation and experiment results in Sec. VI and conclusions in Sec VII.

II. Related Work

Planning and decision strategies for multi-robot systems (and specifically UAV-UGV operation) is a growing topic of interest in robotics research. In [1] the authors present an application to surveillance that takes advantage of the different sensing capabilities of the UAV and UGV. Planning for a heterogeneous system when the UGV acts as a recharging station is examined as a rendezvous schedule problem in [2] and [3], while [4] proposes a control strategy using an artificial force network based on remaining energy.

Much of the research in cooperative multi-robot systems in uncertain environments has been focused on autonomous exploration. Some recent work has focused on exploring environments too large for the robot to hold a standard occupancy map. For example, [5] presents an approach that allows the robot to focus on the boundary (or frontier) regions while [6] and [7] represent the map as a connected graph dense around the robots but sparse for the global map farther away from the robot. To handle scenarios in unknown environments, with unknown task locations, and unreliable communication, [8] proposes a novel approach using epistemic logic to allow the UGV and UAV robots to reason about other robots' beliefs. In these works, the solutions involve allocation of UGVs and UAVs to explore different areas to work in parallel to build a complete map. In contrast, our solution involves the UGV and UAV working directly together to navigate the environment.

There are also efforts to use UAVs to sense different terrains to aid in the path planning for the UGV. The authors in [9] present work that simultaneously plans a route for the UGV while taking into account the terrain classified by a neural network. The work in [10] and [11] use a UAV to collect images of the environment to build an energy-cost map to plan the best path for a UGV. Similar work is presented in [12] for a multi-robot system to plan optimal routing for each robot considering different terrains. Our framework differs from these works by allowing the UAV to explore (while still considering the terrain) simultaneously with UGV navigation, instead of relying on the UAV to first explore the whole environment.

Research for UGV path planning via UAV exploration has also recently received significant interest. A hybrid path planning method is introduced in [13] to: first, use UAV images to help UGV improve recognition of obstacles, and second, use a genetic algorithm to find an optimal path for the UGV. To improve the UAV exploration, the authors in [14] present a method to use frontiers and UGV traversability to reduce the time the UAV needs to find the optimal UGV path. To handle situations where the UGV is unable to detect obstacles, [15] presents a method where the UAV hovers above the UGV during travel with the UAV acting as the only sensor for obstacle detection. Considering safety, the authors of [16] have developed a method for the UAV to scan the route slightly ahead of the UGV to continuously provide updated map information to the UGV. In our work, we expand the capabilities of the UGV-UAV system by allowing the robots to become disconnected, enabling the UAV to explore farther ahead in the environment.

To the best of our knowledge, we believe our work is the first to propose a path planning strategy for a UGV assisted by a UAV that allows both robots to navigate the environment simultaneously while also allowing the two robots to become disconnected, reducing the total navigation time for the UGV while considering real-world constraints found when operating robots in challenging environments.

III. Preliminaries

In this section we provide various assumptions and constraints used to build the proposed planning framework.

Environment: In addition to obstacles, our application considers terrains in the environment. We observe that there are two sub-categories of terrain that occur: 1) *Hazardous* terrain (e.g. pools of water) where the UGV would experience a catastrophic failure if it entered the area and 2) *Rough* terrain (e.g., mud) that can safely be traversed but with degraded performance. For use in planning algorithms, rough terrain is given a constant penalty cost C_t similar to [11], quantifying the efficiency loss of the UGV while travelling over it.

Sensing: Our path planning strategy is generalized to be able to handle various sensing capabilities between the UGV and the UAV. For demonstration purposes, in this paper we assume: 1) both robots are able to detect obstacles in the environments and 2) the UAV is able to detect rough and hazardous terrain (e.g., using a bird's eye view) which the UGV is unable to sense.

Communication: We assume that communication between robots is limited and only possible when the distance between the robots is below some communication radius r_c. Additionally, we consider realistic communication bandwidth limiting the data communicated between the robots to a small number of waypoint coordinates.

System Dynamics: Throughout the paper, we use $\boldsymbol{x}_g = (x_g, y_g)^T$ and $\boldsymbol{x}_a = (x_a, y_a)^T$ to denote the xy position of the UGV and UAV respectively. The explicit system model for the UAV isn't relevant to our framework, however in the case study presented in the paper, we assume the UAV is able to maintain a constant z height and moves in a xy plane. The UGV system dynamics are known and are modeled in discrete time as a function of the current UGV position $\boldsymbol{x}_g$, control input $\boldsymbol{u}_g$, and random disturbances $\boldsymbol{w}$. The variable k is used to signify discrete simulated timesteps. The model,

$$\boldsymbol{x}_g(k+1) = f(\boldsymbol{x}_g(k), \boldsymbol{u}_g(k), \boldsymbol{w}) \tag{1}$$

is used by the UAV to estimate the UGV's trajectory when scouting ahead. In this paper we use the common unicycle model for the UGV, but the proposed method is agnostic to the exact model type.

IV. Problem Formulation

Let us consider a SoS composed of a UGV and a UAV both initialized at the same initial position $\boldsymbol{x}_0$ and tasked to navigate to a desired final position $\boldsymbol{x}_d$ through an uncertain environment while dealing with obstacles, and hazardous or rough terrain (see Preliminaries). We define $\hat{T}_f$ as the total time for the UGV to reach $\boldsymbol{x}_d$ if it was operating completely alone (i.e. without assistance from the UAV scouting ahead). Similarly, let us define T_f^* as the optimal minimum time to reach $\boldsymbol{x}_d$ if the robot had complete knowledge of the environment for the initial plan. Given the aforementioned definitions, our problem can be formally defined as:

Problem: ***System of Systems (SoS) Navigation under Liveness and Safety Constraints***: Derive a path planning policy for the SoS to minimize UGV mission time T_f such that the following boundary conditions for the liveness constraint is respected:

$$T_f^* \leq T_f \leq \hat{T}_f \tag{2}$$

with the goal being that T_f is as close to the ideal time T_f^* as possible. The policy should also guarantee UGV safety by avoiding hazards at all the times:

$$\|\boldsymbol{x}_g(t) - \boldsymbol{h}\| > 0, \ \forall\ t \in [0, T_f], \ \forall\ \boldsymbol{h} \in \mathcal{H} \tag{3}$$

where $\boldsymbol{x}_g(t)$ is the position of the UGV at time t, and $\mathcal{H}$ is the set of all hazards $\boldsymbol{h}$. In the case study focused in this work, the UAV can detect hazards the UGV is unable to detect. Thus, if needed, the UAV should reroute the UGV to a safe path that avoids any hazard.

V. Approach

The proposed framework is a path planning and decision making strategy for a system of systems (SoS) architecture in which a fast and agile UAV augments the sensing capabilities of a slower moving UGV navigating an uncertain environment. Throughout this section, we detail the behavior of each robot in the SoS and the various techniques that combine to create the complete proposed path planning policy.

A. UGV Behavior

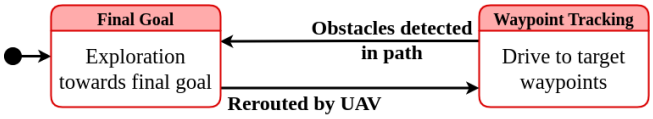


Fig. 2. UGV finite state machine.

The UGV's behavior can be abstracted as the state machine in Fig. 2 with two states each determining the UGV's current target position. Initialized in the *Final Goal* state, the UGV attempts to directly reach the final goal position $\boldsymbol{x}_d$. The UGV remains in this state unless it is rerouted by the UAV which causes a transition to *Waypoint Tracking*. When rerouted, the UAV delivers an ordered set of waypoints $\mathcal{X}_g = \{\boldsymbol{x}_g, \boldsymbol{x}_1, ..., \boldsymbol{x}_d\}$ describing a path avoiding all known obstacles hazards from the current UGV position $\boldsymbol{x}_g$ to the final goal position $\boldsymbol{x}_d$. While in this state, the UGV drives towards each waypoint in order until reaching $\boldsymbol{x}_d$.

If a previously unknown obstacle is detected that crosses the UGV's current path plan, the robot transitions back to the *Final Goal* state, ignoring the waypoints and driving to the final goal position instead. In either mode, the UGV navigates the unknown environment towards the target position using a frontier based method described in the following subsection.

B. UGV Frontier Exploration

For navigation through the unknown environment, the UGV utilizes a frontier-based method [17] for selecting the target position for the current control cycle. The frontier $\mathcal{F}$ is defined as the set of cells in the robot's occupancy map currently marked as *free* cells that are adjacent to an *unknown* cell. By driving to points along the frontier, the robot forces new areas in the environment to be explored.

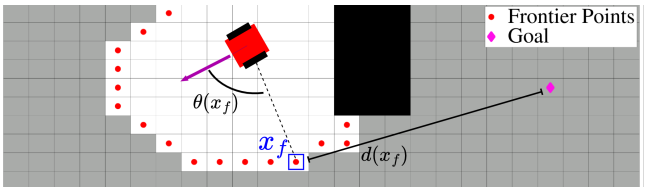


Fig. 3. Frontier points & example cost measures for single frontier point.

When exploring using the frontier method, a single point from the frontier set $\boldsymbol{x}_f^* \subset \mathcal{F}$ is selected to use as the target position. The target point $\boldsymbol{x}_f^*$ is determined by finding the frontier point that minimizes a user defined utility function. For computational simplicity, the cost measures for the utility function used in this work include: 1) a distance cost $d(\boldsymbol{x}_f)$, defined as the point $\boldsymbol{x}_f$'s euclidean distance from the target position and 2) a heading cost $\theta(\boldsymbol{x}_f)$ defined as the angle between the robot's current heading and the line connecting the robot and the point $\boldsymbol{x}_f$. Fig. 3 graphically shows an example of the cost metrics (distance and heading) for one frontier point. By minimizing the distance cost, we encourage exploration towards the target position and by minimizing the heading cost we avoid inefficient switching between different sections of the frontier. The selection of the single point $\boldsymbol{x}_f^*$ from the frontier set is mathematically defined as:

$$\boldsymbol{x}_f^* = \underset{\boldsymbol{x}_f \in \mathcal{F}}{\arg\min} \left(w_d \|C_d(\boldsymbol{x}_f)\| + w_\theta \|C_\theta(\boldsymbol{x}_f)\| \right) \tag{4}$$

where w_d and w_θ are experimentally adjusted weights that determine the importance of each measure to the total cost. The cost components are normalized to avoid comparison of components with different units.

$$\|C_d(\boldsymbol{x}_f)\| = \frac{d(\boldsymbol{x}_f) - d_{min}}{d_{max} - d_{min}} \tag{5}$$

$$\|C_\theta(\boldsymbol{x}_f)\| = \frac{\theta(\boldsymbol{x}_f) - \theta_{min}}{\theta_{max} - \theta_{min}} \tag{6}$$

Unless rerouted by the UAV to follow specific waypoints, the UGV continues to explore using the frontier until it reaches the final goal position.

C. UAV Behavior

An overview of the UAV operation is described here before going into the details in the subsequent subsections. Compared to the UGV, The UAV has a more complex behavior as it has the responsibility of scouting ahead, determining if a better path exists for the UGV, and potentially rerouting the UGV. The finite state machine describing UAV behavior is shown in Fig. 4. In the initial *Particle Follow* state, the UAV explores the environment by tracking the

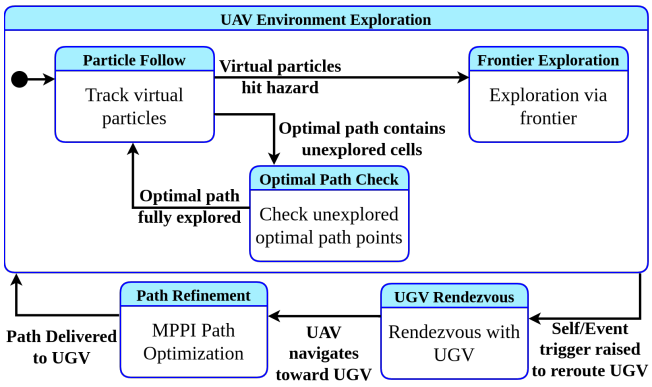


Fig. 4. UAV finite state machine.

predicted UGV trajectory determined from virtual particles representing possible beliefs of future states of the UGV (Sec. V-D). If the UAV discovers hazardous terrain along the predicted UGV trajectory, it transitions to the *Frontier Exploration* state. In this state, the UAV explores using the same frontier method as the UGV (Sec. V-B) until a safe viable path is found for the ground robot. When a potential new path for the UGV is found and the path goes through any cells in the occupancy map currently marked as *unknown*, the UAV transitions to the *Optimal Path Check* state and checks all unexplored cells in the new path to validate the path. Even if a more efficient or safer path for the UGV is determined, our framework encourages more exploration by waiting to deliver the waypoints until absolutely necessary (Sec. V-E). When triggered, the UAV transitions to the *UGV Rendezvous* state to meet and reroute the UGV. In this stage (the *Path Refinement* state), a path optimization procedure for safety using MPPI optimization (Sec. V-F) is run to adjust the new waypoints to be delivered to the UGV. Once delivered, the UAV restarts the scouting and exploration process.

D. Reachability Analysis via Virtual Particles

A necessary component of the work presented in this paper is the ability of the UAV to scout ahead along the predicted trajectory of the UGV. This can be difficult when operating robots in the real world since the prediction procedure needs to handle uncertainties and noise in the system. A typical way to predict future states is with reachability analysis [18], however there are two significant limitations with the technique. First, reachability analysis is computationally expensive [18] and second, standard reachability analysis assumes the robot is controlled using open-loop commands while the UGV in our system continuously uses feedback control. Instead, we propose using a sampling-based prediction policy using a set of virtual particles $\mathcal{P}$ for estimating the reachable set of the UGV. Each particle $\boldsymbol{p}_i \in \mathcal{P}$ represents a possible belief of the UGV prediction. Each particle is a unique simulated instance of the UGV navigating the environment alone. The set $\mathcal{P}$ of n_p total particles is formally defined as,

$$\mathcal{P} = \{\boldsymbol{p}_i = (\boldsymbol{x}_{p_i}, t_p),\ \forall\ i \in [1, ..., n_p]\} \tag{7}$$

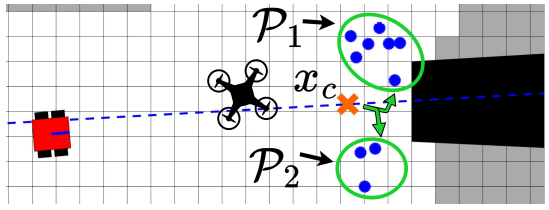
where the ith particle $\boldsymbol{p}_i$ is a tuple containing the simulated position $\boldsymbol{x}_{p_i}$ and associated timestamp t_p of the UGV. Using knowledge of the UGV system dynamics (1) and the control algorithms to determine the input $\boldsymbol{u}_i(k)$, the UAV propagates a particle $\boldsymbol{p}_i$ forward in time by updating its tuple

$$\boldsymbol{x}_{p_i}(k+1) = f(\boldsymbol{x}_{p_i}(k), \boldsymbol{u}_i(k), \boldsymbol{w}) \tag{8}$$

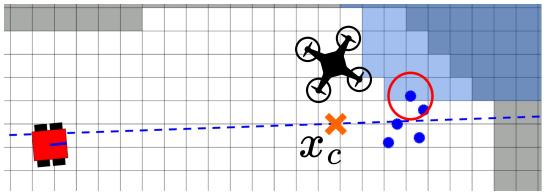
$$t_p(k+1) = t_p(k) + dt \tag{9}$$

where dt is the simulated sample time and $\boldsymbol{w} \sim \mathcal{N}(0, \sigma_g^2)$ is a normal random variable injecting randomness into the prediction to mimic real-world noise and uncertainties. Each control cycle, the UAV repeats (8) and (9) to simulate the particle forward in time until the distance between the particle position $\boldsymbol{x}_{p_i}$ and the UAV position $\boldsymbol{x}_a$ reaches the edge of the UAV sensor range R_a (i.e., $\|\boldsymbol{x}_{p_i} - \boldsymbol{x}_a\| = R_a$). By simulating multiple particles, the sample set $\mathcal{P}$ approximates the distribution of possible trajectories for the physical UGV.

Different than standard reachability analysis, the particle sampling approach allows us to include the UGV's feedback control in the state prediction. By including feedback, the variance of the distribution remains bounded as the control compensates for the random noise and uncertainties, hence clustering all particles together as time progresses. The average position of the particles and corresponding simulation time are used as the estimate of the future position $\hat{\boldsymbol{x}}_g$ at time t_p for the UGV. The values are saved together in a set $\mathcal{V}_g = \{(\hat{\boldsymbol{x}}_g(0), t_p(0)), ..., (\hat{\boldsymbol{x}}_g(k), t_p(k), ..., (\hat{\boldsymbol{x}}_g(N_k), t_p(N_k))\}$ that represents the predicted trajectory for the UGV. The final position $\hat{\boldsymbol{x}}_g(N_k)$ in $\mathcal{V}_g$ represents the farthest point the UGV has currently been simulated. Using $\mathcal{V}_g$ for trajectory prediction of the UGV, the UAV scouts ahead along the trajectory and assess if the UGV could potentially reach unsafe or sub-optimal conditions.



(a) Particle Split



(b) Particle in Hazard

Fig. 5. Virtual particles identifying uncertain/dangerous scenarios for UGV.

A second utility of the virtual particles is to identify two possible issues shown in Fig. 5. In scenario *(a)* a fork in the path is discovered and the UAV is unsure if the UGV would turn left or right, so the UAV doesn't know which direction to scout along. To identify this "particle split" scenario, a set of particle clusters $\mathcal{W}$ is created via a density-based clustering algorithm similar to *dbscan* [19]:

$$\mathcal{W} = \{\ \mathcal{P}_i \mid \bigcup_{i=1}^{N_P} \mathcal{P}_i = \mathcal{P}\ \} \tag{10}$$

$$\mathcal{P}_i = \{\boldsymbol{p}_j \mid \|\boldsymbol{x}_{p_j} - \boldsymbol{x}_{p_l}\| < \delta_c,\ \forall\ \boldsymbol{p}_l \in \mathcal{P}_i\} \tag{11}$$

where each cluster $\mathcal{P}_i$ is composed of particles $\boldsymbol{p}_j$ that are within some distance δ_c from all other particles in the same cluster. The total number of clusters N_P changes during the mission depending on the situation. Fig. 5(a) shows an example with two clusters in $\mathcal{W}$ circled in green. When $N_P > 1$, a "particle split" scenario is identified and the UAV selects the cluster with the largest number of particles to follow. When the UGV approaches the fork, the UAV intentionally reroutes the UGV to take the same path it scouted.

For scenario *(b)* the UGV route lies close to hazardous terrain such that the UGV could potentially drive into an undesired region. To identify this scenario, the UAV checks if any particle position $\boldsymbol{x}_{p_i}$ is within the hazard region as defined by (3). In this case, the UAV marks the current UGV trajectory as unsafe and switches to a frontier exploration mode, treating the hazard as a solid obstacle to avoid, to explore the environment looking for a safe path for the UGV.

In either scenario, the UAV saves the last known safe position of the particles as a *rerouting checkpoint* $\boldsymbol{x}_c$ (shown in Fig. 5), representing the farthest the UGV is allowed to reach before it needs to be rerouted by the UAV.

E. Triggering Procedure for Proactive UGV Reroute

Given the constraint that the UAV and UGV are unable to communicate when apart, a critical piece of our framework is the process to signal the UAV when it should stop exploring and should instead rendezvous with the UGV. We include two types of signal triggers: 1) a self trigger when the UAV has found a better path plan for the UGV; 2) an event trigger when the UGV is approaching a *rerouting checkpoint* $\boldsymbol{x}_c$ (determined from virtual particles as discussed in the previous section). When triggered, the UAV flies towards the UGV to reroute it by delivering an updated set of waypoints $\mathcal{X}^*$. The rest of this section details the necessary algorithms and calculated metrics required for the triggering decision.

1) Rendezvous Point Calculation: While scouting ahead and disconnected from the UGV, the UAV continuously calculates the projected rendezvous point $\boldsymbol{x}_r$ defined as the position the UAV would meet the UGV if it started immediately. Shown in Fig. 6 by the green circled point, the rendezvous point is important for two reasons: 1) The UAV needs to ensure it can reach the UGV to safely reroute it *before* the UGV approaches a *rerouting checkpoint*; 2) The UAV uses $\boldsymbol{x}_r$ as the starting point for calculating the optimal rerouting waypoints described later in this section.

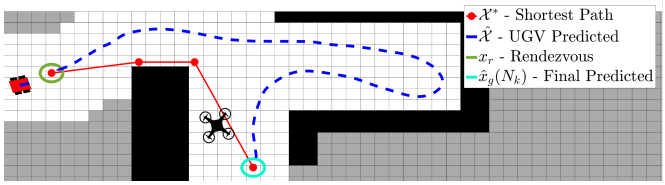


Fig. 6. Comparison of predicted UGV trajectory with optimal path to same position.

The rendezvous point $\boldsymbol{x}_r$ is calculated by the UAV using an iterative approach shown in Algorithm 1. The input includes current position $\boldsymbol{x}_a$ and velocity v_a of the UAV as well as the predicted UGV trajectory $\mathcal{V}_g$ that contains estimated position and corresponding time. Stepping through the predicted UGV trajectory, the algorithm tests how long the UAV would take to fly towards each UGV position. The first point along the trajectory where the UAV could arrive before the UGV is set as the rendezvous point $\boldsymbol{x}_r$.

Algorithm 1 Calculating Rendezvous Point

Input: $\boldsymbol{x}_a$, v_a, $\mathcal{V}_g = (\hat{\boldsymbol{x}}_g(k), \hat{t}(k)) \ \forall \ k$
Output: Rendezvous Point, $\boldsymbol{x}_r$

t_i is the time step to test for meeting point
t_a is time required for UAV
$t_i \leftarrow t_{now}$ ▷ Initialize to current time
while $t_i \leq t_a$ **do**
  $t_i \leftarrow t_i + dt$ ▷ Increment time step
  $\boldsymbol{x}_r \leftarrow \hat{\boldsymbol{x}}_g(t_i)$ ▷ Rendezvous point to test
  $d_a \leftarrow \|\boldsymbol{x}_a - \boldsymbol{x}_r\|$ ▷ Distance from UAV
  $t_a \leftarrow d_a / v_a$ ▷ UAV flight time
return $\boldsymbol{x}_r$

2) UAV Trigger Signal Procedure: With the rendezvous point $\boldsymbol{x}_r$ and farthest point in the predicted trajectory $\hat{\boldsymbol{x}}_g(N_k)$, an optimal path can be calculated and compared against the current UGV path (see Fig. 6). Although many methods exist to determine an optimal path, in this work we choose to use a standard A^* algorithm with a post-processing smoothing algorithm based on line-of-sight to reduce the number of points. The heuristic for a given position $\boldsymbol{x}$ used by our A^* algorithm is,

$$H(\boldsymbol{x}) = 1/C_t \cdot \|\boldsymbol{x} - \hat{\boldsymbol{x}}_g(N_k)\| \tag{12}$$

where $\hat{\boldsymbol{x}}_g(N_k)$ is the end of trajectory position and C_t is a terrain efficiency cost mentioned in Sec. III. For nominal conditions, C_t equals 1 and (12) equals the distance to the goal. For rough terrain (e.g. mud or sand), $C_t < 1$ based on the efficiency reduction the UGV experiences when traversing the terrain. For ease of discussion in this work, we assume the terrain cost value C_t is known by the UAV for all detected terrain.

The output from the path planning algorithm is a series of waypoints $\mathcal{X}^*$ that define the optimal path for the UGV. The optimal path is compared against the predicted UGV path $\hat{\mathcal{X}}$ (the path the UGV *would* take if it doesn't receive any assistance from the UAV) determined by the position elements of $\mathcal{V}_g$. Intuitively, if there is a significant difference between the $\hat{\mathcal{X}}$ and $\mathcal{X}^*$, then it is advantageous to reroute the UGV to follow $\mathcal{X}^*$. Fig. 6 shows an example instance where the UAV has found a more efficient path for the UGV and self-triggers to rendezvous with the UGV to reroute.

To mathematically compare $\hat{\mathcal{X}}$ and $\mathcal{X}^*$, a utility function $S_X(\cdot)$ is used to quantify the cost of a path. The details of the utility function can be application specific but an obvious example would be UGV total travel time. With $S_X(\cdot)$, we define a difference metric Δ_X to quantify the improvement of $\mathcal{X}^*$ over $\hat{\mathcal{X}}$:

$$\Delta_X = S_X(\hat{\mathcal{X}}) - S_X(\mathcal{X}^*) \tag{13}$$

Using Δ_X, the self-triggering event for UAV is raised when the following conditions are both satisfied:

$$\Delta_X > \delta_X \tag{14}$$

$$\Delta_X < \Delta_{max} - \delta_m \tag{15}$$

where δ_X is a user defined value to prohibit the UAV wasting time flying back for only a minimal gain. Δ_{max} is defined as the maximum Δ_X observed so far and δ_m is a small threshold to handle noise. If the UGV starts to travel down the less optimal path, the improvement from re-routing will begin to decrease, thus the value of Δ_X will also decrease. Condition (15) is used to enable the UAV exploring as long as possible while triggering to deliver new waypoints to the UGV at the optimal time. As an example,

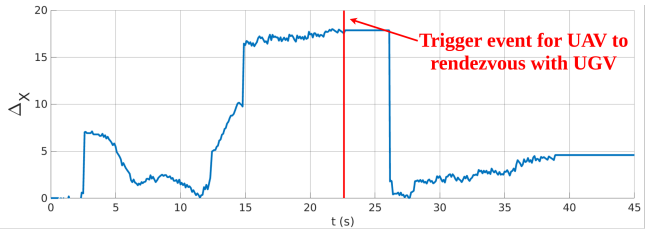


Fig. 7. Difference metric for test case in Fig. 1.

Fig. 7 shows the computed Δ_X metric for the experiment in Fig. 1. While flying ahead and exploring the environment, the UAV discovers the path is blocked (at $t = 12$s) and Δ_X quickly grows, signifying a significant difference between the predicted and optimal paths. The self-triggering event is raised at $t = 22.5$s to signal the UAV to rendezvous with the UGV and reroute it.

The triggering procedure also needs to handle the scenarios in which the UGV approaches a hazard or a fork in the path. In order to meet the UGV as it reaches the rerouting checkpoint $\boldsymbol{x}_c$, the event trigger signal is raised on the UAV when the distance between the rerouting checkpoint $\boldsymbol{x}_c$ is within some distance threshold δ_t of the rendezvous position $\boldsymbol{x}_r$, formally when $\|\boldsymbol{x}_c - \boldsymbol{x}_r\| < \delta_t$. When triggered by the potential fork case, the UGV is rerouted along the same path the UAV has already explored.

F. MPPI Based Waypoint Modification

Once triggered, the UAV delivers a new set of waypoints to the UGV. However, due to the nonholonomic design of the robot, noise, and other non-linearities of the system, the UGV will not exactly follow the straight line path described by the waypoints. As a result, the waypoints may potentially lead to a dangerous trajectory for the ground robot. For example, driving alongside a boundary of a hazard could result in catastrophic failure with even slight deviations.

To deal with this issue, our framework uses a model predictive path integral (MPPI) optimization procedure to modify the UGV target waypoints before delivering to the UGV. Due to real-time constraints, it would be infeasible to run the optimization for the full path through the environment. Instead, the MPPI operation optimizes the UGV trajectory over a finite time horizon $t \in [t_0, t_0 + T_H]$. Because the procedure is sampling based and iterative, the computation can be time-bound to ensure completion when the UAV is within communication range of the UGV.

As shown in our previous work [20], the MPPI procedure for optimizing waypoints can be formulated as a stochastic optimal control problem:

$$\mathcal{X}' = \arg\min_{\mathcal{X}} \mathbb{E}\left[J(\mathcal{X} + \mathcal{E})\right] \tag{16}$$

where $\mathcal{X}$ is a fixed set of target waypoints for the UGV, $\mathcal{E} = \{\epsilon_x, \epsilon_y\}$ is a random variable adding xy perturbation, and $J(\cdot)$ is a cost function whose expected value is minimized. For this framework, the waypoints from the optimal path planner $\mathcal{X}^*$ are used as the initial waypoints $\mathcal{X}$ for (16). With the perturbed waypoints $\mathcal{X} + \mathcal{E}$ and known starting position $\boldsymbol{x}(t_0)$, the UGV is simulated to generate sample trajectories.

The cost function is a combination of three components:

$$J(\mathcal{X}) = C_F(\mathcal{X}) + C_{obs}(\mathcal{X}) + C_r(\mathcal{X}) \tag{17}$$

Where the terminal cost $C_F(\mathcal{X})$ is defined as the distance from the final position $\boldsymbol{x}(T_H)$ in the simulated sample trajectory to the final waypoint $\boldsymbol{x}_F$ in $\mathcal{X} + \mathcal{E}$, multiplied by a weighting factor α_f:

$$C_F(\mathcal{X}) = \alpha_f \|\boldsymbol{x}(T_H) - \boldsymbol{x}_F\| \tag{18}$$

The second term $C_{obs}(\mathcal{X})$ is an obstacle cost that penalizes any trajectory that intersects with an occupied cell. The term is composed of a weighting factor α_o multiplied by an indicator function that has a value of 1 if the trajectory collides with an obstacle and is 0 otherwise.

$$C_{obs}(\mathcal{X}) = \alpha_o I_o(\mathcal{X} + \mathcal{E}) \tag{19}$$

Finally, $C_r(\mathcal{X})$ is the running cost of the trajectory that takes into account changes in terrain.

$$C_r(\mathcal{X}) = \alpha_r \sum_{k=1}^{T_H} \rho(\boldsymbol{x}(t_k)) \tag{20}$$

For our application we define the running cost function $\rho(\cdot)$ whose value depends on the terrain at $\boldsymbol{x}(t_k)$. For rough terrain, the cost is the inverse to the terrain efficiency loss C_t for the terrain type at position $\boldsymbol{x}$. For hazards, a large constant δ_H is selected to heavily penalize a trajectory that crosses hazards, as defined by (3).

$$\rho(\boldsymbol{x}) = \begin{cases} \delta_H & \text{if } \boldsymbol{x} \text{ in hazard} \\ 1/C_t & \text{if } \boldsymbol{x} \text{ rough terrain} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

With the cost function defined, we use (17) to get costs $\mathcal{C} = \{c_1, c_2, ..., c_N\}$ for N total sample trajectories of $\mathcal{X}+\mathcal{E}$. We can then follow the procedure developed in [20] to approximate the solution $\mathcal{X}'$ to (16). When completed, the modified waypoints $\mathcal{X}'$ overwrite $\mathcal{X}^*$ as the waypoints to deliver to the UGV.

VI. Results

In this section we present simulation and experiment results to validate the proposed SoS framework on a UGV assisted by a UAV in unknown and hazardous environments.

A. Simulations

Simulations were conducted to validate the strategy in 20 different environments. Some environments were created using randomized obstacle placement while others were manually designed. Several environments also included hazards and/or rough terrain. The UGV was modeled as a nonholonomic differential drive robot with linear and rotational velocity control inputs $\boldsymbol{u}_g = [v, w]^T$. Flying in a raised xy-plane, the UAV control input consists of a linear velocity

command divided into xy-components $\boldsymbol{u}_a = [v_y, v_y]$. We allowed the UAV to move at most 3 times faster than the UGV, with sensor range 50% larger than the UGV's.

For comparison, we ran four different scenarios for each case: 1) *Proposed*: the UAV-UGV SoS using our proposed strategy, 2) *UGV Alone*: The UGV navigates alone without UAV guidance, 3) *Ideal*: The true optimal where the complete environment is known, and 4) *UAV Explore*: The UGV waits to drive until the UAV explores enough to find the true optimal path (similar in principle to [9] or [10]). Shown in Fig. 8, the mission completion time results were recorded, or marked as a failure if the UGV drove into a hazard. In each case our proposed strategy improved the total time compared to a *UGV Alone* or *UAV Explore* scenarios. When needed, the UAV successfully rerouted the UGV away from a hazard. Thus both the liveness and safety guarantees are validated.

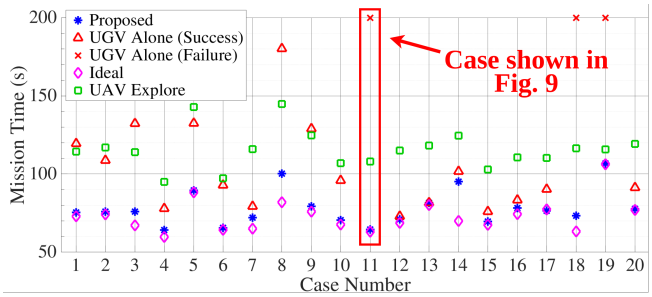


Fig. 8. Simulation mission completion times comparison for 20 different environments containing obstacles, rough terrain and hazards. Videos available at *https://www.bezzorobotics.com/ps-iros24*

In Fig. 9, we show snapshots from one case using our framework. Fig. 9(a) shows the initial setup for the environment containing obstacles depicted in dark grey, hazards (e.g. pools of water) in blue, and rough terrain (e.g. mud) in pink. For this simulation, all obstacles, terrain and hazard are initially unknown. In Fig. 9(b) the UAV, while scouting ahead, detects particles running into the first hazard. Since the UGV is still far away, the UAV continues to explore using the frontier method. The UAV discovers that the gap between the two hazards is blocked, so it searches for a feasible path. Once found, the UAV flies back to rendezvous with the UGV to proactively reroute the ground robot, as shown in Fig. 9(c). Fig. 9(d) shows the waypoints calculated from the A* algorithm result in a path close to a hazard. Since multiple particles drive into the hazard, the UAV determines that this path is unsafe for the UGV and adjusts the waypoints. In Fig. 9(e), the target waypoints are adjusted by the MPPI procedure to get a safe trajectory around the hazard. Finally, Fig. 9(f) shows the UGV reaching the goal position. We highlight that in this case the UAV decided to reroute the UGV through a shorter section of the rough terrain instead of the straight line path shown in Fig. 9(d) or a path that completely avoids the rough terrain.

B. Experiments

Our framework was also validated with laboratory experiments using a Husarion ROSbot and Bitcraze Crazyflie quadrotor. Position data was measured using Vicon motion capture system. The UGV's LiDAR range was limited to 1m while the UAV sensor range was artificially set to 1.6m and was allowed to move 4 times faster than the UGV. Experiments were conducted in several different configurations of obstacles and hazards. The experiments were run with the multi-robot system running our planning framework and with the UGV operating alone as a comparison.

In each setup, our proposed strategy successfully rerouted the UGV to a shorter or safer path to reach the final goal position. In the experiment shown at the beginning of this paper in Fig. 1, the UAV successfully rerouted the UGV to a more efficient path to help the UGV reduce total travel time by 37% in comparison to the UGV-alone case. In Fig. 10, we show results from an experiment with a hazard directly in the path of the UGV. In Fig. 10(a), the UAV has started to scout ahead and detects the hazard. The UAV continues to explore until triggered to rendezvous when the UGV is getting close to the hazard, Fig. 10(b). In Fig. 10(c), the UAV rendezvous with the UGV and reroutes it. After delivering the safe plan, the UAV restarts the scouting process until reaching the final goal position, Fig. 10(d). Determining that the UGV is already on an efficient and safe path, the UAV lands and the mission finishes when the UGV reaches the final goal position, Fig. 10(e). As a comparison, Fig. 10(f) shows the scenario when the UGV is unassisted: not able to detect the hazard, the UGV drives straight ahead until falling into the hazard, failing the mission.

VII. Conclusion

In this paper we presented a novel strategy to unify a heterogeneous multi-robot system with a System of Systems (SoS) abstraction. Using a UGV-UAV pair of robots as an example, we propose a planning strategy taking advantage of the different sensing and dynamic abilities of the robots. By predicting the slower UGV trajectory, the faster UAV is able to scout ahead in the unknown environment and determine if the current UGV path plan is inefficient or unsafe and should be rerouted with an updated path plan. Using a self-triggering process, the UAV is able to delay rendezvous with the UGV to continue exploration. We validated our work both in simulations and experimentally in the lab.

For future work, we plan to generalize our framework for different robot combinations beyond a UGV-UAV. This includes working with multi-robot systems larger than 2 robots or different robot types such as legged robots and UAV-UAV combinations. Additionally, we hope to deploy the framework in more realistic settings by conducting outdoor experiments.

VIII. Acknowledgements

This work is based on research sponsored by Northrop Grumman through the University Basic Research Program

References

[1] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

[2] G. Shi, N. Karapetyan, A. B. Asghar, J.-P. Reddinger, J. Dotterweich, J. Humann, and P. Tokekar, "Risk-aware uav-ugv rendezvous with chance-constrained markov decision process," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 180–187.

[3] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[4] T. X. Lin, E. Yel, and N. Bezzo, "Energy-aware persistent control of heterogeneous robotic systems," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 2782–2787.

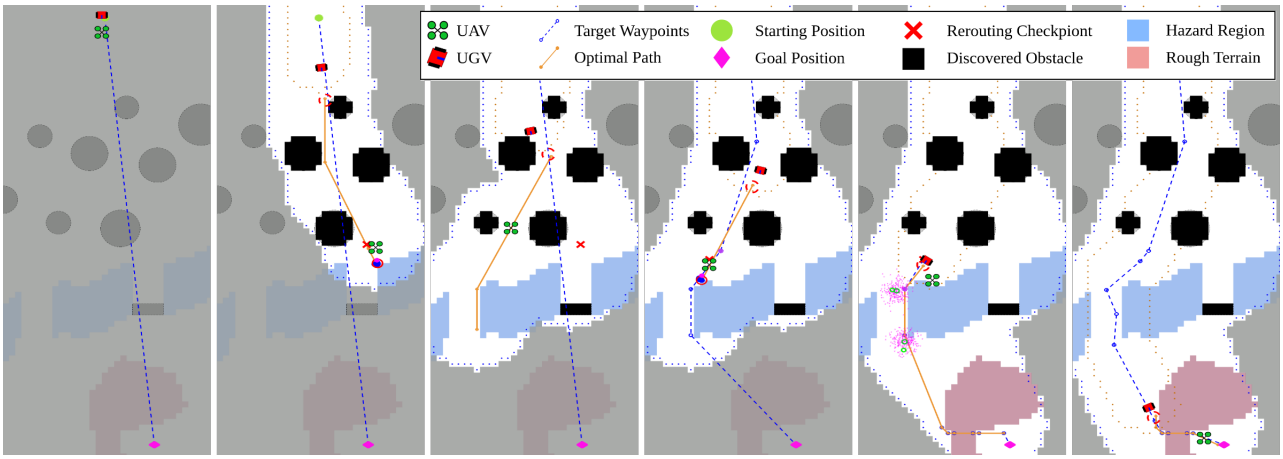


Fig. 9. Snapshots of simulated case study showing UAV rerouting UGV to final goal position while safely avoiding hazard.

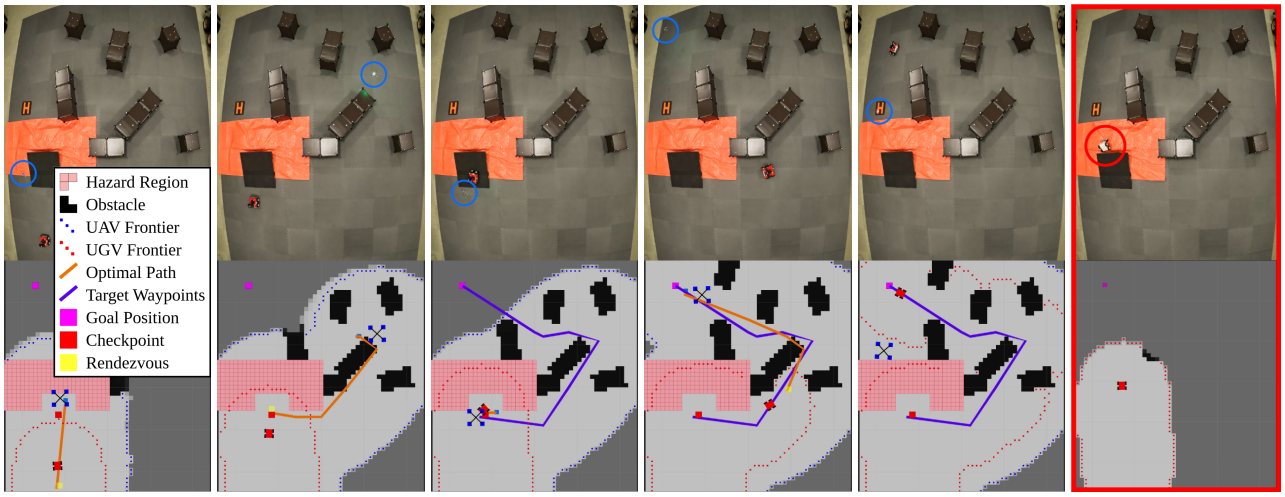


Fig. 10. (a)-(e) Snapshots and corresponding RViz visualizations from single lab experiment [UAV circled in blue] running proposed framework to help UGV avoid hazard. (f) Same setup results in UGV falling into hazard when operating without UAV.

[5] J. Williams, S. Jiang, M. O'Brien, G. Wagner, E. Hernandez, M. Cox, A. Pitt, R. Arkin, and N. Hudson, "Online 3d frontier-based ugv and uav exploration using direct point cloud visibility," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2020, pp. 263–270.

[6] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, R. Siegwart, M. Hutter, and K. Alexis, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3306–3313.

[7] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020.

[8] L. Bramblett and N. Bezzo, "Epistemic planning for heterogeneous robotic systems," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 691–698.

[9] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.

[10] M. Wei and V. Isler, "Air to ground collaboration for energy-efficient path planning for ground robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1949–1954.

[11] B. Sofman, J. A. Bagnell, A. Stentz, and N. Vandapel, "Terrain classification from aerial data to support ground vehicle navigation," *Carnegie Mellon University*, 2006. [Online]. Available: https://doi.org/10.1184/R1/6561173.v1

[12] J. Banfi, A. Messing, C. Kroninger, E. Stump, S. Hutchinson, and N. Roy, "Hierarchical planning for heterogeneous multi-robot routing problems via learned subteam performance," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4464–4471, 2022.

[13] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A hybrid path planning method in unmanned air/ground vehicle (uav/ugv) cooperative systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9585–9596, 2016.

[14] S. Zhang, X. Zhang, T. Li, J. Yuan, and Y. Fang, "Fast active aerial exploration for traversable path finding of ground robots in unknown environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[15] C. Chen, Y. Wan, B. Li, C. Wang, G. Xie, and H. Jiang, "Motion planning for heterogeneous unmanned systems under partial observation from uav," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1474–1479.

[16] B. Gilhuly and S. L. Smith, "Robotic coverage for continuous mapping ahead of a moving vehicle," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 8224–8229.

[17] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang, and Y. W. Yun, "An improved frontier-based approach for autonomous exploration," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 292–297.

[18] E. Yel, T. X. Lin, and N. Bezzo, "Computation-aware adaptive planning and scheduling for safe unmanned airborne operations," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 575–596, 2020.

[19] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Trans. Database Syst.*, vol. 42, no. 3, jul 2017.

[20] J. Higgins, N. Mohammad, and N. Bezzo, "A model predictive path integral method for fast, proactive, and uncertainty-aware uav planning in cluttered environments," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.