Terrain Topology-Informed Motion Planning for Tactical UGVs

Jyotirmoy Mukherjee[1], Giri M. Kumar[1], Hannah White[1], Andrea L'Afflitto[2] and Corina Sandu[1]

Abstract— **This paper presents a novel guidance system for autonomous unmanned ground vehicles (UGVs) operating in unknown, potentially contested environments. This system applies to UGVs relying on onboard sensors only and operating in GNSS-denied environments since it only requires a point cloud map of the environment, which is formed as the vehicle proceeds. Furthermore, the location and threat capabilities of the opponents, if any, are unknown. The proposed guidance system enables those classical tactics being put in place by ground forces and law enforcement agents operating in the presence of potential threats. Numerical simulations in highly realistic point cloud maps display the applicability of the proposed results.**

I. INTRODUCTION

In recent years, there has been a growing demand from law enforcement agencies to support human operators with autonomous mobile robots, such as autonomous unmanned ground vehicles (UGVs). These vehicles may provide logistic support and help explore unknown, potentially hostile outdoor or indoor environments. Thus, these vehicles lower the potential risk for human operators and may provide intelligence, which is vital to plan successive operations. To be successful, these UGVs need to be able to operate tactically, that is, covertly reduce the chances of being identified, targeted, and disabled by the opponents.

This paper proposes a guidance system for autonomous UGVs operating tactically, that is, in covertly, in unknown, potentially contested environments. This system does not require any external source of information, such as a GNSS (global navigation satellite system) signal, such as GPS or Galileo, but only necessitates a point cloud map generated, for instance, by onboard cameras or LiDAR systems. A unique feature of the proposed guidance system lies in the fact that the opponents' location and threat capabilities are unknown and can not be discovered. Thus, the proposed system can not be cast within a deterministic of stochastic game-theoretic approach [1]–[3]. The proposed guidance system enables well-assessed tactics already employed by human operators in potentially contested environments, such as coasting obstacles to seek shelter and exploiting speed as the distance from safe areas increases. Additional tactics include moving faster as the distance from areas deemed safe, such as the perimeter of obstacles, increases. This set of tactics is far from being comprehensive, but it suffices to show the proposed design philosophy. Other approaches, such as maximizing the distance traveled within areas of the point cloud map observed and classified by the navigation system with a higher degree of confidence, are not considered, but can be seamlessly implemented in the proposed framework.

This work ports some of the concepts presented in [4]–[6], which has pioneered the design of guidance systems for unmanned aerial vehicles (UAVs) operating in contested environments without any knowledge of the opponents' location and capabilities, to UGVs. Although both the proposed guidance system for UGVs and the guidance system for UAVs examined in [4]–[6] have ontological overlaps, their implementations differ for multiple significant reasons. Firstly, in the case of UAVs, any obstacle in a point cloud map must be avoided, whereas in the case of UGVs, obstacles in the point cloud map must be classified into terrain and actual obstacles. Although obstacles must be avoided at all times, terrain points must be leveraged to plan the UGV's path on them. Furthermore, point cloud maps are usually relatively sparse. In planning paths and trajectories for UAVs, this sparsity is accounted for by voxels of suitable dimensions that cluster those portions of the space containing a sufficiently large number of points of the cloud corresponding to objects. To plan paths for UGVs on point clouds representing terrains, the proposed guidance system approximates the terrain features using a Delaunay triangulation of points of the cloud classified as part of the terrain, and includes the incenters of the Delaunay triangles in the terrain map. The proposed guidance system exploits a trajectory planning algorithm based on the use of splines, which is significantly faster than the approach presented in [4]–[6], which requires the solution of a highly dimensional convex optimization problem in some neighborhoods of the vehicle. Finally, the planned attitude, which is essential to determine one of the UGV's control inputs, namely the steering angle, is determined by mean of a rotation-minimizing frame. Similar to the Frénet-Serret reference frame, a rotation-minimizing reference frame comprises the unit tangent vector, which serves as reference roll axis of the UGV. The other two unit vectors are deduced from the normal and binormal unit vectors of the Frénet-Serret reference frame, while minimizing the reference frame's rotation about the roll axis. Thus, the use of a rotation-minimizing frame allows reducing a classical problem of the Frénet-Serret reference frame, namely sudden variations in the direction of the normal and binormal unit vectors at inflection points of the reference trajectory. In [4]–[6], the attitude planning problem is solved through a convex optimization algorithm applied to the feedback-linearized equations of motion of the UAV.

[1]J. Mukherjee, G. M. Kumar, H. White and C. Sandu are with Virginia Tech, Department of Mechanical Engineering, Blacksburg, VA 24061, USA

[2]A. L'Afflitto is with the Grado Department of Industrial and Systems Engineering of Virginia Tech, Blacksburg, VA, USA (E-mail: a.lafflitto@vt.edu)

Although the tactical guidance problem for mobile ground robots in contested environments has already been explored, this field is at its earliest stages. Worthy of mention is CoverNav [7], where a deep reinforcement learning method is being employed to plan coverted trajectories for off-road UGVs. In particular, CoverNAP selects obstacles as shelters to hide behind thus overriding the segmentation problem employed in this paper and enabling the use of some user-defined criteria for the choice of the sheltering obstacle. The Authors in [8] present a solution to the path planning problem in contested environments, wherein the opponents' locations are unknown, by exploiting the terrain elevation. In [9], a potential field method is employed to design tactical paths assuming that the traversable spaces are at the same elevation and obstacles are the only means of maintaining covertness. Worthy of mention is that potential field methods usually produce local solutions since the presence of multiple obstacles may create local minima. None of these works seem to emphasize the relevance of planning a path or a trajectory that accounts for the terrain features and, hence, the cost of its traversability.

This work is structured as follows. In Section II, we outline the proposed solution to the path planning problem for UGVs in contested environments. Specifically, given a user-defined goal set, we present a heuristics-based approach to the tactical guidance problem that involves planning sequences of waypoints on terrain points identified by the onboard guidance system. In this section, we also discuss how the point cloud map produce by the onboard navigation system is partitioned into terrain and obstacle sets. In Section III, we discuss the proposed solution to the trajectory planning problem. This approach is based on the use of cubic splines while enforcing a constant velocity along the trajectory on each spline segment or sets of consecutive spline segments. Section IV outlines the attitude planning problem that, as already discussed, is essential to determine the UGV's steering angle. In Section V, we highlight the features of the proposed guidance system through numerical simulations. These simulations have been performed in a highly realistic point cloud map of an outdoor environment formatted and sampled as if it were generated by commercial-off-the-shelf depth and tracking cameras. Finally, in Section VI, we draw conclusions and outline future work directions.

II. Path Planner

A. Overview

In this section, we present a path planner for autonomous UGVs operating in potentially contested environments. Given a point cloud map generated by some onboard system, such as cameras, the proposed path planner outlines a sequence of waypoints from the UGV's current location to a user-defined *goal set* $\mathcal{G}$, while avoiding obstacles, and while minimizing the vehicle's exposure to potential threats. In this paper, we assume that the location and capabilities of opponents are unknown, and, hence, the path planner must enable several known tactics such as coasting obstacles and avoiding unknown areas to reduce the exposure to opponents.

To describe the proposed path planner, let $\mathbb{I} \triangleq \{O; X, Y, Z\}$ represent a gravity-aligned orthonormal reference frame centered at O, with Z positive upwards. The environment $\mathcal{C} \subset \mathbb{R}^3$ detected by the onboard navigation system is partitioned into two subsets, namely the *non-traversable set*, $\mathcal{F}_c \subset \mathcal{C}$, which can not be penetrated by the UGV, and its complement to the configuration space $\mathcal{C}$, known as the *traversable set* $\mathcal{F}$. In this research, the non-traversable set $\mathcal{F}_c$ is considered to be time-invariant, that is, once a non-traversable area is detected, then its topology does not change in time. However, the map of the environment is unknown *a priori* as is disclosed as the field of view of the vehicle's sensors scan $\mathcal{C}$. In this work, we consider the unexplored space, that is, the space not yet detected by the onboard sensors, as traversable.

B. Obstacle-terrain segmentation and mapping

The point cloud generated by the navigation system, which represents the non-traversable set $\mathcal{F}_c$, comprises both obstacles and the terrain. Consequently $\mathcal{O}$ necessitates segmentation to separate the points belonging to terrain, on which planning must be performed, and obstacles, which need to be avoided by the UGV. This segmentation is achieved through a RAndom SAmple Consensus (RANSAC) algorithm [10]. RANSAC robustly segments plane surfaces in point cloud data by selecting random subsets of points iteratively, and estimating a plane model and a number of inliers in each iteration. The algorithm returns the "best-fit-plane" that corresponds to the largest number of inliers; for details, see Algorithm 1.

Let $\mathbb{J} \triangleq \{O'; X', Y', Z'\}$ be an orthonormal reference system centered at $O' \in \mathcal{C}$. Let $n \in \mathbb{R}^3$ denote the normal vector to the plane produced by the RANSAC algorithm, which is within a specified cone angle about the UGV's Z' axis. A point cloud $p \in \mathcal{F}_c$ is an inlier if its distance from the plane, denoted by $\eta \geq 0$, is such that $\eta \leq \epsilon$, with $\epsilon > 0$ being a user-defined distance threshold. The points tagged as "inliers" collectively represent the terrain $\mathcal{T}$, whereas the remaining point clouds are considered as obstacle points $\mathcal{O}$ such that $\mathcal{T} \cup \mathcal{O} = \mathcal{F}_c$. Once classified, the points in $\mathcal{F}_c$ are scan aligned, that is, recorded in the global reference frame $\mathbb{I}$ and registered in a grid structure known as the *voxel map* in $\mathbb{I}$. The resolution of the voxel map is chosen to be compatible with the size of the UGV. Thus, the mapping system provides the *terrain map*

$$\mathcal{M}_{\mathcal{T}} \triangleq \{(i, j, v_{i,j}) \mid (x_i, y_j) \in \mathcal{T}, i, j \in \mathbb{N}, v_{i,j} = 0\}, \quad (1)$$

where $v_{i,j} = 0$ indicates that the point $(x_i, y_j, z_{i,j}) \in \mathcal{M}_{\mathcal{F}_c}$ is a terrain voxel and $v_{i,j} = 1$ indicates that $(x_i, y_j, z_{i,j})$ is an occupied voxel.

Whereas the terrain topology in the real world manifests as a continuous shape, $\mathcal{M}_{\mathcal{T}}$ is modeled as a discrete grid structure. This discretization, while computationally convenient, falls short of effectively capturing the nuanced curvature inherent to the terrain's topology. To ameliorate this discrepancy, a mesh comprising triangular elements is superimposed over $\mathcal{T}$, facilitating a closer approximation

Algorithm 1 RANSAC based point cloud segmentation

Input: Point cloud $\mathcal{F}_c$, cone angle threshold ϵ_ϕ, distance threshold ϵ

Output: Terrain set $\mathcal{T}$ & obstacle set $\mathcal{O}$

function RANSACSEGMENT($\mathcal{F}_c$, ϵ_ϕ, ϵ)
  $\mathcal{T} \leftarrow \varnothing$, $\mathcal{O} \leftarrow \varnothing$, $\mathcal{I}_{\max} \leftarrow$ null, n_{iter} ▷ Initialize variables
  while $k < n_{iter}$ **do**
    $\mathcal{S} \leftarrow$ Random subset of points of $\mathcal{F}_c$
    $\mathbf{P} \leftarrow$ Fit plane to $\mathcal{S}$
    $\mathcal{I} \leftarrow$ points in $\mathcal{S}$ s.t. $\|\mathbf{P} - \mathcal{S}\| \leq \epsilon$ & $\cos^{-1}(\langle \hat{n}, Z' \rangle) \leq \epsilon_\phi$
    if $|\mathcal{I}| > |\mathcal{I}_{\max}|$ **then**
      $\mathcal{I}_{\max} \leftarrow \mathcal{I}$
    end if
  end while
  $\mathcal{T} \leftarrow \mathcal{I}_{\max}$
  $\mathcal{O} \leftarrow \mathcal{F}_c \setminus \mathcal{T}$
end function

to the actual terrain topology. The Delaunay triangulation method [11] is employed for this purpose since it maximizes the smallest angle in each of the triangles within the triangulation, thereby mitigating the occurrence of overly elongated triangles. In the context of Delaunay triangulation, the incenters of each triangular element are incorporated into $\mathcal{T}$ and the corresponding map $\mathcal{M}_\mathcal{T}$. The updated map $\mathcal{M}_\mathcal{T}$ facilitates the generation of topology-informed waypoints in complex terrains.

C. Tactical path planning

The path planning algorithm is predicated on the A-star (A*) graph search algorithm. The underlying cost function is defined as

$$f_k \triangleq g_k + h_k, \quad k \in \{0, \ldots, n_p\}, \tag{2}$$

where

$$g_k \triangleq \sum_{q=0}^{k} [\kappa(d_2(\hat{r}_q, \mathcal{O})) d_2(\hat{r}_q, \hat{r}_{q-1})] \tag{3}$$

denotes the *cost-to-come function*, $\hat{r}_0 \in \mathcal{T}$ denotes the UGV's current location and $\{r_k\}_1^q \subset \mathcal{T}$ denotes the UGV's subsequent planned waypoints, $d_2(\cdot,\cdot)$ denotes the *Euclidean distance*,

$$h_k \triangleq (1 - \mu_2) d_2(\hat{r}_k, \mathcal{G}) \tag{4}$$

denotes the *heuristic function* and

$$\kappa(\alpha) \triangleq 1 - \mu_2 e^{4\mu_1\mu_3 - (\mu_3\alpha + \mu_1\alpha^{-1})^2}, \qquad \alpha > 0. \tag{5}$$

The weighing function $\kappa : (0, \infty) \to (0, 1]$ is introduced to imbue the path planner with tactical behavior. A path is considered more tactical if it coasts the obstacle set $\mathcal{O}$ as it reduces the UGV's exposure to detection as well as external threats and reckless otherwise. The parameters μ_1, μ_2, and μ_3 can be adjusted by the user to modulate the algorithm's propensity towards reckless or tactical behavior. It has been demonstrated in [4] that the modified heuristic function is both admissible and consistent, thereby ensuring the optimality of the reference path.

The user-defined parameters μ_1, μ_2, and μ_3 influence the path's *op-tempo*, dictating the algorithm's inclination towards rapid goal attainment or strategic obstacle circumnavigation. The parameter, μ_2 exerts a scaling effect on the heuristic function, directly affecting the urgency of goal pursuit. Specifically, setting $\mu_2 = 0$ produces the least coverted, hence most reckless, trajectory, whereas larger values of μ_2 produce more tactical results. The propensity towards seeking shelter on the other hand is dictated by the $\kappa(\cdot)$ function which scales the *cost-to-come*. The function $\kappa(\cdot)$ produces a bias in the cost-to-come function attracting the path toward the goal set. The minimizer of $\kappa(\cdot)$ is $\sqrt{\mu_1\mu_3^{-1}}$ [4]. Consequently, the choice of μ_1 and μ_3 affects the path behavior. The prescribed ranges for the parameters are established to be $\mu_1 \in (1, \infty)$, $\mu_2 \in [0, 0.5]$, and $\mu_3 \in (0.7, \infty)$ for reckless behaviors and $\mu_1 \in (0, 1)$, $\mu_2 \in (0.5, 1]$, $\mu_3 \in (0, 0.7)$ for tactical behavior. For an in-depth exploration of the effects of the tunable parameters and the weighing function, readers are directed to [4]. Future work directions involve modifying (3) so that the reference path maximizes the distance traveled within regions of $\mathcal{T}$ with classified by the onboard navigation system at a higher confidence level.

III. Trajectory Planner

A. Overview

Having defined the reference path, a reference trajectory is defined as a piecewise function of time interpolating consecutive subsequences of the reference path. This process thereby not only dictates the precise *tempus* of traversal but also "smoothens" the path. In this research, the trajectory is designed using a cubic polynomial spline due to its attribute of exhibiting $C^2([0, \infty))$ continuity. Furthermore, the cubic spline is parameterized by a monotonically increasing variable, which enables the regulation of velocity and acceleration.

B. Cubic spline planning

Consider the piecewise continuous cubic spline trajectory represented by the $r : \mathbb{R} \to \mathbb{R}^3$, where

$$r(\xi) \triangleq [x(\xi), y(\xi), z(\xi)]^{\mathrm{T}}, \tag{6}$$

$\xi \in [0, \infty)$ denotes a spatial parameter. For each $\xi \in [0, \infty)$, the components $x(\xi)$, $y(\xi)$, and $z(\xi)$ of the interpolating spline segment are defined as

$$x(\xi) = \sum_{i=0}^{3} a_i \xi^i, \tag{7}$$

$$y(\xi) = \sum_{i=0}^{3} b_i \xi^i, \tag{8}$$

$$z(\xi) = \sum_{i=0}^{3} c_i \xi^i, \tag{9}$$

where $a_i, b_i, c_i \in \mathbb{R}$. The determination of the spline coefficients involves solving a tri-diagonal linear system predicated on Hermite interpolation [12], ensuring continuity and smoothness of the spline trajectory. After the acquisition of these coefficients, the spatial variable ξ is parameterized to maintain a constant velocity profile along the interpolating path.

C. Spline parameterization with constant velocity

The *arc length* of the piecewise continuous cubic spline is given by

$$L(\xi) = \int_0^\xi \|r'(\zeta)\| \, \mathrm{d}\zeta, \quad \xi \in [0, \infty), \tag{10}$$

where

$$\|r'(\xi)\| \triangleq \sqrt{x'(\xi)^2 + y'(\xi)^2 + z'(\xi)^2}. \tag{11}$$

Imposing a user-defined constant traversal velocity $V > 0$, it must hold that

$$V = \frac{\mathrm{d}L(\xi(t))}{\mathrm{d}t} = \frac{\partial L(\xi)}{\partial \xi} \frac{\mathrm{d}\xi(t)}{\mathrm{d}t}, \qquad t \in [0, \infty). \tag{12}$$

Employing the Leibniz rule for integration, we find that

$$\frac{\partial L(\xi)}{\partial \xi} = \|r'(\xi)\|, \tag{13}$$

and, hence, it follows from (12) that

$$V = \|r'(\xi(t))\| \frac{\mathrm{d}\xi(t)}{\mathrm{d}t}, \tag{14}$$

$$\frac{\mathrm{d}\xi(t)}{\mathrm{d}t} = \frac{V}{\|r'(\xi)\|}. \tag{15}$$

Since V is constant, $\xi(\cdot)$ is computed as

$$\xi(t) = \int_0^t \frac{V}{\|r'(\xi(\tau))\|} \mathrm{d}\tau, \qquad t \in [0, \infty). \tag{16}$$

In this paper, the translational velocity V can be varied across consecutive spline segments to enforce different tactical behaviors. For instance, let $\overline{\xi} \in [0, \infty)$ denote the lower extremum of a spline segment. Then, over that spline segment, we set

$$V = \mathrm{sat}\left(\eta \, d_2(\overline{\xi}, \mathcal{O}), V_{\min}, V_{\max}\right), \tag{17}$$

where $\eta > 0$ is user-defined, $V_{\max} \geq V_{\min} > 0$ are user-defined, and $\mathrm{sat}(V, V_{\min}, V_{\max}) \triangleq \min\{\max\{V, V_{\min}\}, V_{\max}\}$ denotes the *saturation function*. This way, the UGV's speed increases with its distance from the obstacle set, which is considered safe to coast, within some user-defined interval $[V_{\min}, V_{\max}]$.

IV. Attitude Planner

The cubic spline $r(\xi(t))$, $t \in [0, \infty)$, encapsulates the spatial and temporal evolution of the UGV. Determining the orientation of the UGV along the polynomial trajectory becomes imperative to derive reference steering angles for controller tracking. Let ${}^{\mathbb{I}}R(\xi)^{\mathbb{J}} \in SO(3)$, $\xi \in [0, \infty)$, denote the rotation matrix that maps the UGV reference frame $\mathbb{J}$ to the global frame $\mathbb{I}$. The Euler angle triplet derived from the rotation matrix at any point ξ on the spline gives the UGV orientation at that point. The three columns of ${}^{\mathbb{I}}R(\xi)^{\mathbb{J}}$ can be constructed as the unit tangent, normal, and binormal vectors to $r(\xi)$ for all $\xi \in [0, \infty)$.

Since the cubic polynomial (6) is a twice differentiable curve in $\mathbb{R}^3$, parameterized by ξ, the unit *tangent vector* along the spline is defined as

$$T(\xi) \triangleq \left\| \frac{\mathrm{d}r(\xi)}{\mathrm{d}\xi} \right\|^{-1} \frac{\mathrm{d}r(\xi)}{\mathrm{d}\xi}, \qquad \xi \in [0, \infty). \tag{18}$$

The tangent vector denotes the UGV's instantaneous heading in $\mathbb{R}^3$. It is worthwhile noting that, since the UGV's translational velocity along the spline is a non-zero constant, (18) is well-defined for all $t \in [0, \infty)$ and, hence, for all $r(\xi(t))$. The unit *normal vector* is defined as

$$N(\xi) \triangleq \frac{1}{\gamma(\xi)} \frac{\mathrm{d}T(\xi)}{\mathrm{d}\xi}, \qquad \xi \in [0, \infty), \tag{19}$$

where

$$\gamma(\xi) \triangleq \left\| \frac{\mathrm{d}T(\xi)}{\mathrm{d}\xi} \right\| \tag{20}$$

denotes the *curvature* of $r(\cdot)$. It can be observed that $\langle T(\xi), N(\xi) \rangle = 0$ for all $\xi \in [0, \infty)$, that is, $T(\xi)$ and $N(\xi)$ are perpendicular vectors. Finally, the binormal vector is defined as

$$B(\xi) \triangleq T^{\times}(\xi) N(\xi), \qquad \xi \in [0, \infty), \tag{21}$$

where $(\cdot)^{\times}$ denotes the *cross product operator*.

The set $\{r(\xi), T(\xi), N(\xi), B(\xi)\}$ forms the Frénet-Serret frame centered at $r(\xi)$, $\xi \in [0, \infty)$ [13]. It can be observed that, at inflection points, the curvature, γ approaches zero and the curve transitions from bending in one direction to the opposite. Consequently, a Frénet-Serret frame undergoes a reorientation about the tangent vector $T(\xi)$, which makes it impractical to set ${}^{\mathbb{I}}R(\xi)^{\mathbb{J}} = [T(\xi), N(\xi), B(\xi)]$. To overcome this issue, in this paper we define ${}^{\mathbb{I}}R(\cdot)^{\mathbb{J}}$ employing a method that stems from the Frénet-Serret reference frame, namely we employ a rotation-minimizing reference frame.

Rotation-minimizing frames are orthonormal reference frames that minimize the *torsion*

$$\tau(\xi) = \frac{\langle (r'(\xi))^{\times} r''(\xi), r'''(\xi) \rangle}{\| (r'(\xi))^{\times} r''(\xi) \|^2}, \qquad \xi \in [0, \infty), \tag{22}$$

which can be interpreted as a measure of the speed of rotation of the binormal vector at the given point. Thus, similarly to a Frénet-Serret reference frame, a rotation-minimizing frame comprises the unit tangent vector as the first axis. Thus, the remaining two axes of a rotation-minimizing frame undergo

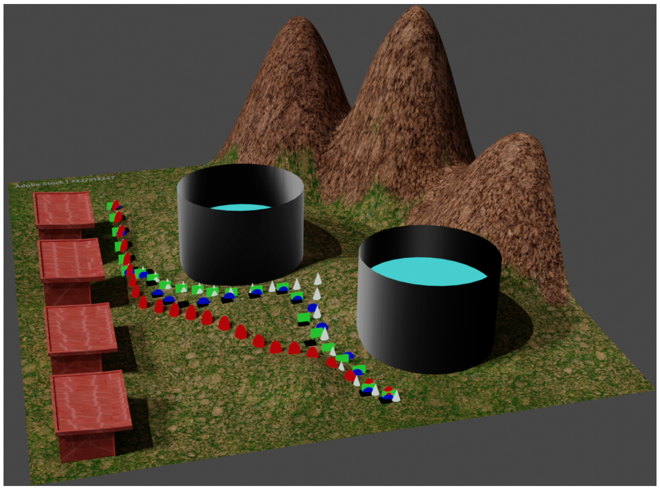

Fig. 1. Paths for 5 different mission scenarios. All missions require the UGV to leave from the same initial position and reach the goal set. The only difference between missions lies in the parameters μ_1, μ_2, and μ_3, which have been set as in Table I. It is apparent how more coverted, and, hence, more tactical paths, coast obstacles to a greater extent, while more reckless trajectories aim to the goal set more directly

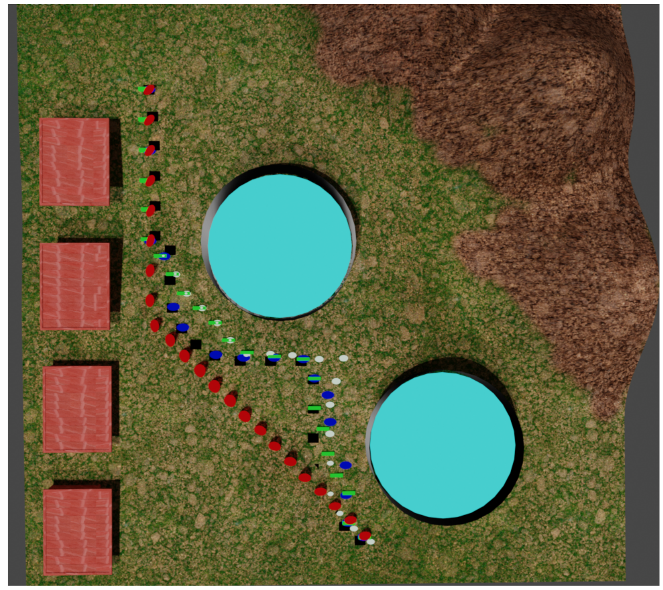

Fig. 2. Top-down view of the paths shown in Figure 1

parallel transport along the curve. In this paper, we employ the approach presented in [14] to deduce these two axes, and, hence, deduce ${}^{\mathbb{I}}R(\xi)^{\mathbb{J}}$, $\xi \in [0, \infty)$, and the angles of an underlying 3-2-1 rotation sequence; for details, see Algorithm 2.

Algorithm 2 Rotation-minimizing frame generation [14]

Input: Position on spline $\{r(\xi_i)\}_{i=0}^{n}$, initial right-handed reference frame $\mathbb{J}(\xi_0) = \{r(\xi_0); T(\xi_0), N(\xi_0), B(\xi_0)\}$

Output: Right-handed reference frames $\mathbb{J}(\xi_i) \triangleq \{r(\xi_i); T(\xi_i), N(\xi_i), B(\xi_i)\}$, $i = 0, \ldots, n$

function RMF($\mathbb{J}(\xi_0), r(\xi_i)$)
  while $r(\xi_i) \notin \mathcal{G}$ **do**
    $v_1 \leftarrow r(\xi_{i+1}) - r(\xi_i)$
    $c_1 \leftarrow \langle v_1, v_1 \rangle$
    $B^L(\xi_i) \leftarrow B(\xi_i) - (2v_1/c_1)\langle v_1, B(\xi_i)\rangle$
    $T^L(\xi_i) \leftarrow T(\xi_i) - (2v_1/c_1)\langle v_1, T(\xi_i)\rangle$
    $v_2 \leftarrow T(\xi_{i+1}) - T^L(\xi_i)$
    $c_2 \leftarrow \langle v_2, v_2 \rangle$
    $B(\xi_{i+1}) \leftarrow B^L(\xi_i) - (2v_2/c_2)\langle v_2, B^L(\xi_i)\rangle$
    $N(\xi_{i+1}) \leftarrow B^{\times}(\xi_{i+1})T(\xi_{i+1})$
    $\mathbb{J}(\xi_{i+1}) \leftarrow \{r(\xi_{i+1}); T(\xi_{i+1}), N(\xi_{i+1}), B(\xi_{i+1})\}$
  end while
end function

V. Numerical simulations

The guidance system detailed in this paper has been tested in a virtual environment generated using Blender [15] and simulations were run on an Intel Core i7-9750H CPU. This virtual environment, extending over 500 m $\times$ 500 m, emulates the irregular terrain characteristics of an off-road area, complete with strategically distributed obstacles. Empirical results accentuate the algorithm's capability in pathfinding amidst intricate topographical features, thus signifying its transferability to real-world navigational applications.

The UGV is tasked with reaching the goal set $\mathcal{G}$ from the starting point $\hat{r}_0$. The goal set is shaped as a disk of radius 5 m and located at $[376, 184, 0]^{\mathrm{T}}$ relative to $\hat{r}_0$. Five scenarios were scrutinized, each predicated setting μ_1, μ_2, μ_3 as in Table I. In the most tactical mission (Mission 5), whose waypoints are denoted by the white waypoints in Figure 1, the algorithm coasted the obstacle set $\mathcal{O}$ more closely, thereby diminishing its detectability and vulnerability. In contrast, the most reckless mission (Mission 1), whose waypoints are denoted by the red waypoints in Figure 1, resulted in the most direct path to the goal set. Missions 2, 3, and 4 show increasing levels of covertness by coasting obstacles more and more closely. Table I shows the Euclidean distance traveled over each mission scenario, and it is apparent how less tactical missions also lead to shorter distances traveled.

Figure 3 shows both the torsion and the curvature of the reference frame $\mathbb{J}(\cdot)$ applying the classical Frénet-Serret approach and the rotation-minimizing frame approach to attitude planning. It is apparent how the rotation-minimizing frame approach to attitude planning induces a considerably smaller torsion. Furthermore, numerical evidence, omitted for brevity, clearly show how, applying the rotation-minimizing frame, the UGV's reference pitch and roll axes are consistent throughout the reference trajectory. Applying the Frénet-Serret approach, instead, the UGV's reference pitch and roll axes experience sudden variations at inflection points of the reference trajectory, which make this approach unsuitable to deduce the UGV's steering angle.

VI. Conclusions

This paper presented a guidance system for autonomous UGVs operating tactically in unknown, potentially contested environments. Exclusively leveraging a point cloud map,

TABLE I

COMPARISON OF THE LENGTH OF TRAVERSAL FOR VARYING LEVELS OF RECKLESSNESS AND COVERTNESS

Mission	μ_1	μ_2	μ_3	Path length
1	2.00	0.00	2.00	456 m
2	1.50	0.25	1.00	507 m
3	0.10	0.50	0.50	492 m
4	0.50	0.60	0.10	509 m
5	1.00	0.90	0.70	516 m

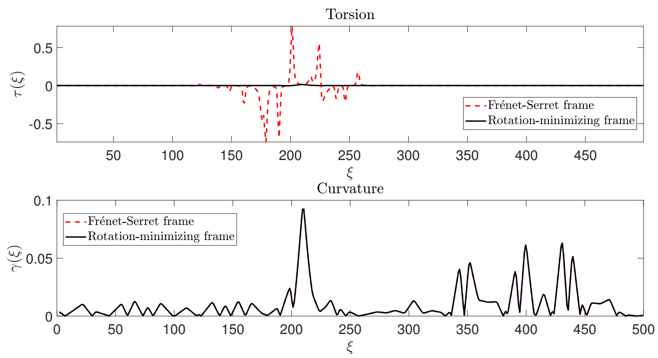

Fig. 3. Torsion and curvature of reference frames. The Frénet-Serret reference frame experiences considerably larger variations in its torsion, which leads to sudden changes in the direction of the UGV's pitch and yaw axes

which can be generated by onboard sensors, this guidance system outlines a reference path, a reference trajectory, and a reference attitude for a mobile ground robot tasked with reaching a given goal set while reducing its risk of exposure to external threats. Since the threats' locations and capabilities are unknown, this guidance system enables several tactics such as coasting obstacles to protect one flank and exploiting speed to seek cover as soon as possible.

Several features distinguish this guidance system. Worthy of mention is that the proposed guidance system plans the vehicle's reference path on some approximation of the terrain. This approximation of the terrain features is produced from the point cloud map by separating actual obstacles from terrain points using a RANSAC algorithm. Successively, the Delaunay triangulation allows adding more data points in the form of incenters of each triangle. The trajectory planner is fast since it relies on cubit splines and global. The use of splines assures continuous differentiability of the reference trajectory with respect to time as well as the arc length. Numerous existing trajectory planners do account for the vehicle's dynamics, but produce results in some convex neighborhoods of the vehicle or are significantly more expensive because based on some non-convex optimization technique. The attitude planner, which is needed to determine the UGV's steering angle, produces a rotation-minimizing reference frame. This reference frame, and, hence, the associated orthonormal matrix, are similar to the reference frame and the orthonormal matrix underlying the classical Frénet-Serret reference frame. However, despite the Frénet-Serret reference frame, the rotation-minimizing reference frame is not prone to rotations about the UGV's roll axis due to changes in the curvature of the reference trajectory.

Future work directions involve several aspects of this work. For instance, the use of machine learning and artificial intelligence methods was avoided to allow low computational costs and be implementable on fast cars equipped with relatively inexpensive computers. However, the use of semantics would allow choosing what obstacles would provide a safer shelter. Furthermore, information on the terrain properties gathered by onboard cameras as well as other proprioceptive sensors can be used to estimate the vehicle's maximum speed and turn radius to plan more realistic trajectories. Finally, test drives would ultimately allow verifying the applicability of the proposed approach.

REFERENCES

[1] S. LaValle and J. Hinrichsen, "Visibility-based pursuit-evasion: the case of curved environments," in *International Conference on Robotics and Automation*, vol. 3, pp. 1677–1682, IEEE, 1999.

[2] R. Murrieta-Cid, H. Gonzalez-Banos, and B. Tovar, "A reactive motion planner to maintain visibility of unpredictable targets," in *International Conference on Robotics and Automation*, vol. 4, pp. 4242–4248, IEEE, 2002.

[3] K. Mann, "Evolving robot behavior to play hide and seek," *Journal of Computational Science in Colleges*, vol. 18, no. 5, p. 257–258, 2003.

[4] J. A. Marshall, R. B. Anderson, W.-Y. Chien, E. N. Johnson, and A. L'Afflitto, "A guidance system for tactical autonomous unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 103, pp. 1–36, 2021.

[5] J. A. Marshall, P. Binder, and A. L'Afflitto, "An optimization-based guidance system for tactical multi-rotor uavs mapping contested environments," in *Conference on Control Technology and Applications*, (Bridgetown, Barbados), pp. 249–254, IEEE, 2023.

[6] J. A. Marshall, P. Binder, and A. L'Afflitto, *Autonomous Multi-rotor Unmanned Aerial Vehicles for Tactical Coverage*, pp. 31–76. Switzerland: Springer, 2024.

[7] J. Hossain, A. Faridee, N. Roy, A. Basak, and D. E. Asher, "Covernav: Cover following navigation planning in unstructured outdoor environment with deep reinforcement learning," in *International Conference on Autonomic Computing and Self-Organizing Systems*, (Los Alamitos, CA), pp. 127–132, IEEE, 2023.

[8] D. Page, A. Koschan, M. Abidi, and J. Overholt, "Ridge-valley path planning for 3D terrains," in *International Conference on Robotics and Automation*, pp. 119–124, IEEE, 2006.

[9] E. Birgersson, A. Howard, and G. Sukhatme, "Towards stealthy behaviors," in *International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1703–1708, 2003.

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[11] F. Aurenhammer, R. Klein, and D.-T. Lee, *Voronoi diagrams and Delaunay triangulations*. World Scientific Publishing Company, 2013.

[12] L. Schumaker, *Spline Functions: Computational Methods*. Philadelphia, PA: SIAM, 2015.

[13] F. Miller, A. Vandome, and J. McBrewster, *Frénet-Serret Formulas: Vector Calculus, Curve, Derivative, Euclidean Space, Kinematics, Darboux Frame, Differential Geometry of Curves, Affine Geometry of Curves*. New York, NY: Alphascript Publishing, 2010.

[14] W. Wang, B. Jüttler, D. Zheng, and Y. Liu, "Computation of rotation minimizing frames," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 1, pp. 1–18, 2008.

[15] Blender online community, *Blender: A 3D modelling and rendering package*. Blender foundation, Stichting Blender Foundation, Amsterdam, 2018.