On Autonomous Ground Vehicle Navigation in Leader-Follower Off-road Environments

Miloš Prágr[1] [0000−0002−8213−893X], Jan Feber[1] [0000−0001−6546−7187], Roman Adámek[2] [0000−0002−8026−5740], Vsevolod Hulchuk[1] [0009−0003−8809−3052], Jan Mazal[2] [0000−0001−5741−558X], and Jan Faigl[1] [0000−0002−6193−0792]

[1] Faculty of Electrical Engineering, Czech Technical University in Prague,
Technická, 2, 166 27, Prague, Czechia
{pragrmi1,feberja1,hulchvse,faiglj}@fel.cvut.cz
https://comrob.fel.cvut.cz

[2] Faculty of Military Robotics, University of Defence, 662 10 Brno, Czechia
{roman.adamek,jan.mazal}@unob.cz

Abstract. In this paper, we report on the developed solution for the autonomous leader-follower system, where an uncrewed ground vehicle follows a human-driven vehicle in outdoor field deployments. The solution consists of the leader vehicle detection and tracking, planning the follower's path, and autonomous navigation of the follower along the leader's path, considering its traversability capabilities using a locally built map. The follower runs its localization system to align 3D scans of its surrounding environment into the map used for path planning. The leader detection is based on the combined LiDAR and RGB perception, where LiDAR data provide near-field object detection, including estimates of its distance and orientation. On the other hand, the RGB image is used to estimate the direction of the detected object. Relatively bumpy terrain causes frequent loss of line of sight between the leader and follower vehicles. Therefore, the tracking is supported by the motion model of the leader vehicle, and its detected positions are tracked using an extended Kalman filter. The tracking is further supported by the pose estimate of the follower and projecting the leader's positions into a local map frame. The localization is a GNSS-free stand-alone solution based on modified LIO-SAM using LiDAR sensors, which has been shown to be sufficient. The developed solution has been deployed in an outdoor field testing site where it successfully performs leader following for about 1 km long trail with median velocity about $1\,\mathrm{m\,s^{-1}}$.

1 Introduction

The studied problem is motivated by the Loyal Wingman scenario [7,11], where uncrewed vehicles follow a leader vehicle that can be human-crewed or uncrewed. In our study case, we consider human-driven leader vehicles traveling in field environments on dirt roads, followed by an autonomous uncrewed vehicle. The leader-follower is a relatively well-studied problem [15], especially within the context of convoying [2,19], using methods such as visual servoing [3]. The basic concept is to detect the leader vehicle using range sensors, such as cameras or LiDARs [12], and steer the follower in the direction of the leader [9]. However, a particular deployment needs parameterization and customization of the existing methods to the specific scenario setup, such as considering possible differences in traversability capabilities of the vehicles in offroad environments [16].

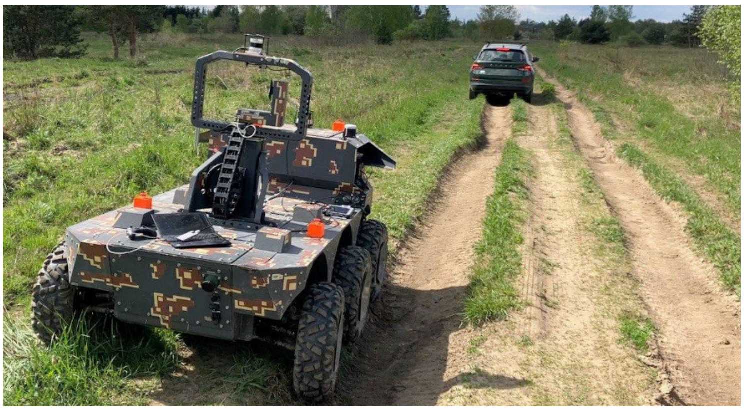


Fig. 1. Leader-follower system in operation.

The proposed solution is built on the existing methods and deployed using the TAROS *Uncrewed Ground Vehicle* (UGV) as the follower and the Škoda Kodiaq crewed leader vehicle, both depicted in Fig. 1. The developed leader detection is a combination of the LiDAR-based detection module based on *Cascade Attention* (CasA) network for 3D object detection [20], initially used as the detection module of the best LiDAR-only tracker in the KITTI multi-object tracking benchmark [6]. It has been further combined with RGB camera images with leader vehicle segmented using YOLOv8 [8]. Positions of the Camera-LiDAR-based detected leader vehicle are tracked using *Extended Kalman Filter* (EKF) tracker coupled to a detection-trajectory matching schema [21]. The tracked leader positions are treated as a path to be followed using a local spline-based planner that can be used with respect to (w.r.t.) to the obstacle grid map created from the LiDAR scans aligned to a point cloud map using *Simulation Localization and Mapping* (SLAM) based on the modified LIO-SAM [14].

The developed solution has been deployed within a military compound of the 44th Light Motorized Batallion, Czech Army, located in Jindřichův Hradec, Czechia [10]. The system performance has been evaluated with the different maximum throttle of the follower vehicle with maximal achieved median velocity up to about $1.5\,\mathrm{m\,s^{-1}}$. The developed solution proved to be viable, and we achieved fully autonomous deployment on the trail, which is about 1 km long, without any human intervention at median velocity approx. $1\,\mathrm{m\,s^{-1}}$.

The rest of the paper is organized as follows. The system setup and overview of the navigation stack used are presented in the following section. A brief description of the used localization is provided in Section 3. The developed leader detection and tracking are summarized in Section 4. Results of the field deployment are reported in Section 5. Finally, concluding remarks are in Section 6.

2 System Setup

The developed leader-follower solution consists of a crew-operated leader vehicle and a UGV follower that is tasked to detect and follow the leader. The crewed leader is a Škoda Kodiaq, and the autonomous follower is the TAROS v1 UGV. The system solution is developed within the navigation stack supporting autonomy [4], and the particular implementation originates from [5]. The overview of the vehicles, used sensors, and utilized existing modules of the stack are presented in the rest of this section. The main components developed, which consist of localization and leader detection and tracking, are presented in the following sections.

2.1 Vehicles and Sensors Used

The leader vehicle used is the *Škoda Kodiaq* depicted in Fig. 2a. The vehicle size is about $4.69 \times 1.88 \times 1.65\,\mathrm{m}$. Although it can achieve high speeds on roads, a safe velocity over bumpy terrains has been experienced up to $20\,\mathrm{km\,h^{-1}}$ (approx. $6\,\mathrm{m\,s^{-1}}$).

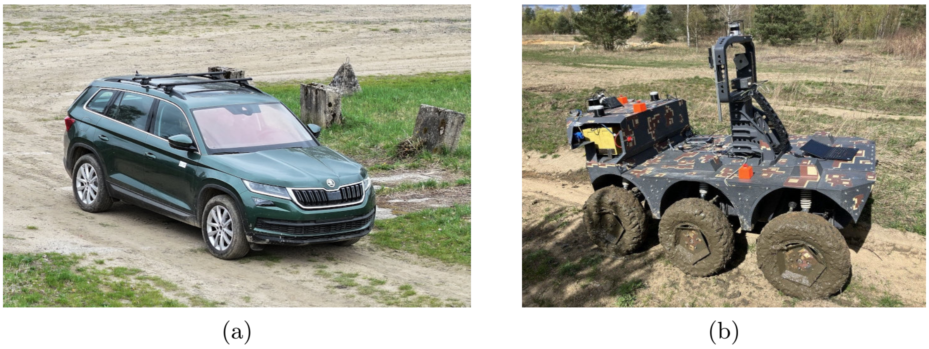

(a) (b)

Fig. 2. (a) Škoda Kodiaq as a crewed leader vehicle; and (b) TAROS UGV used as the follower.

The follower is the TAROS platform by VOP CZ, s.p. [18], a six-wheeled vehicle with a weight of about 1.2 t. The platform size is about $2.74 \times 1.77 \times 2.04\,\mathrm{m}$ with $2.00 \times 0.80\,\mathrm{m}$ wheelbase. Its ground clearance is $0.23\,\mathrm{m}$, and its safe pitch and roll are 10°. The vehicle's maximum operational speed is approx. $6\,\mathrm{m\,s^{-1}}$. However, due to the bumpy environment and previous experience with mud accumulation at higher speeds, the vehicle was limited to $3\,\mathrm{m\,s^{-1}}$ during the autonomous operation, and the vehicle has been mostly operated at lower speeds about $1\,\mathrm{m\,s^{-1}}$. The vehicle with the used sensors is depicted in Fig. 2b.

The UGV can be operated in three modes. In the *Circle* mode, front and rear wheels are steered to achieve high maneuverability. In the *Tatra* mode[3], the front wheels are steered fully, and the middle wheels are steered partially to achieve stability at speed over bumpy terrain. In the *Crab* mode, all wheels are steered to enable diagonal motion. Given the bumpy nature of the traversed circuit, the *Tatra* mode was used for the majority of the experiments.

The primary sensory payload includes a top-mounted *Ouster OS1-128* LiDAR sensor operating in 1024 × 10 mode and a front-mounted *Basler Ace 2 a2A1920-51gcPRO* RGB camera. A secondary front-mounted *Ouster OS2-128* LiDAR is used for long-range sensing, operating in 4096 × 5 mode. The LiDAR is primarily used to detect the leader vehicle using a LiDAR-based detector. The robot also carries *Novatel PwrPak7* GNSS (GPS in particular) unit, and in selected deployments, a *3DM-GX5-AHRS MicroStrain* IMU has been recorded. However, the GNSS unit, regardless of its RTK capability, has not been used, and we solely rely on LiDAR-based localization, further overviewed in Section 3.

The follower's computational resources are based on the workstation with the *Intel Core i7-12700F Alder Lake* 4.9 GHz accompanied with 32 GB DDR5 memory, *NVIDIA GeForce RTX 3070* 8 GB GPU, and 1000 GB SSD. The system is running *Ubuntu 20.04 LTS* [17] and ROS Noetic [13], with a possible extension to ROS2 via a ROS bridge.

2.2 Navigation Stack

The navigation stack is based on the stack originating from [5] that has been developed as the Autonomy part of [4] to integrate perception, global planning, local planning, and control. The perception module of the original stack is designed to process the UGV's sensory data. It provides the detected obstacles and leader position, vehicle position, and terrain model in the terrain type segmentation and occupancy. However, the available implementation comprises only the occupancy model, and therefore, the stack has been adopted by new perception modules that include leader detection, follower localization, and leader tracking. The overview of the stack is depicted in Fig. 3.

The used environment representation is an obstacle grid map with the cell size $\boldsymbol{d_\nu} = 0.5\,\mathrm{m}$, which is refreshed at 10 Hz with max point age 0.5 s. Hence, components such as leader detection and tracking and vehicle localization are replaced by available modules, with some of them used in [1]. Hence, the map is cleared every second. The map is fixed w.r.t. the mission starting position, and its maximal size has been set to 200 × 200 m, which covers the whole deployment area.

[3] Named after steering of Tatra trucks.

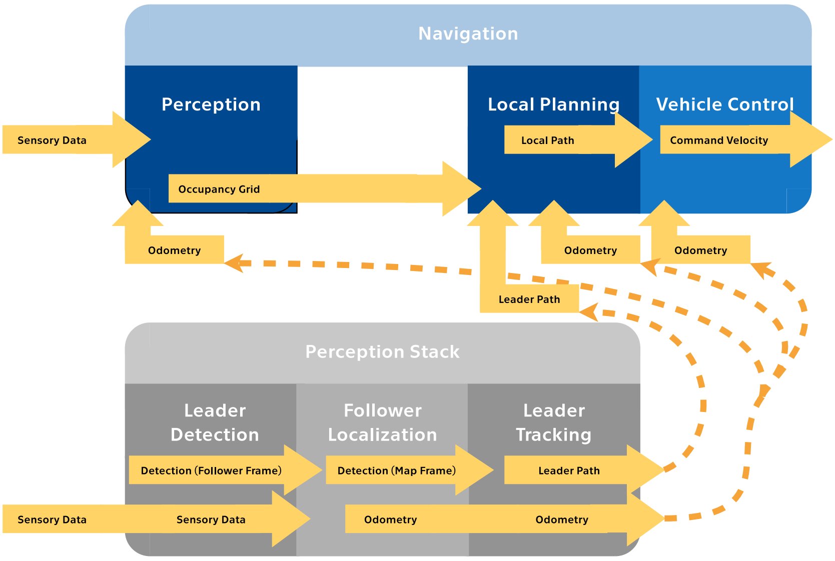
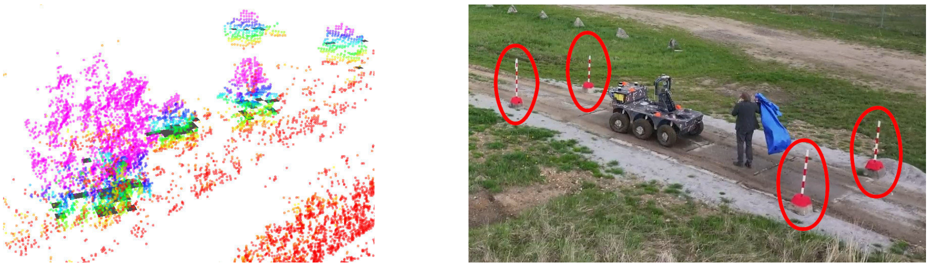


Fig. 3. Overview of the navigation stack used.



(a) Grid map

(b) Concrete pillars

Fig. 4. (a) Obstacles cells (black squares) determined in the obstacle grid map overlaid over LiDAR point cloud (color-coded elevation, red low, purple high). (b) Concrete pillars in the red ellipses on the road are not considered obstacles if the tolerance threshold ignores bumps in the road and small, traversable vegetation.

The leader is followed by the follower relatively tightly (about 10 m), which causes the leader to be detected as an obstacle by the stack that may interfere with the local planner if an obstacle lingers in the map for too long. Therefore, the obstacle grid was further adjusted to disregard obstacles detected in areas where the leader was recently detected. An example of the obstacle cells is depicted in Fig. 4a.

During the field tests, several configurations of obstacle detection sensitivity were considered in the form of cell slope threshold $s_{\max}$. Low (sensitive) values of $s_{\max} = 0.5\,\mathrm{rad}$ discriminate both grass and some bumpy areas encountered during the field tests as obstacles, despite the robot having no issues traversing them, while $\boldsymbol{s}_{\max} = 1.0\,\mathrm{rad}$ detects only large obstacles such as bushes and trees, as depicted in Fig. 4a. While enlarging the obstacle tolerance does turn such obstacles traversable, it hinders the robot from discriminating against small obstacles such as the concrete pillars depicted in Fig. 4b. Besides, since the considered leader-follower formation is following in leader tracks, an alternative solution is to assume the path traversed by the leader is traversable and thus disregard the obstacle detection though setting $\boldsymbol{s}_{\max} = 20.0$.

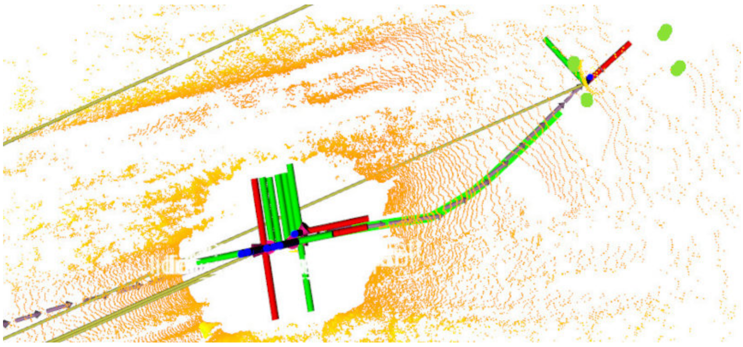


Fig. 5. The path planned by the spline planner (in green) along the reported leader path (in purple) is overlaid over the LiDAR point cloud.

The planning component consists of the global and local planners. The global planner subscribes to the perception modules and a given set of waypoints and reports a global path using the A* algorithm. In the leader-follower deployments, we consider relatively tight following along the detected leader's path; the detected leader path is coupled directly to the local planner. The local planning is facilitated by the baseline spline planner that navigates the robot along the leader-follower reported waypoints w.r.t. the obstacle grid as visualized in Fig. 5. The planner is configured with 20 m lookahead and 1 rad max steering angle. The planned local path is reported to the control component.

The control component drives the UGV along the path determined by the local planner. The velocity command w.r.t. the Pure Pursuit and PID is sent to the controllers that are configured for the follower role of the vehicle. The controller configuration was adapted to drive the TAROS UGV with the max throttle (throttle coefficient) $3\,\mathrm{m\,s^{-1}}$, steering coefficient 3.5, max steering angle 30°, and wheelbase 1.4 m. The PID proportional, integral, and derivative components are set to 0.05, 0.2, and 0.1, respectively. The controllers are set up to account for the relatively slow wheel angle change of the UGV and to eliminate the zig-zag behavior that is a common issue for the slow-turning vehicle. The configuration results in overall median velocity approx. $1.5\,\mathrm{m\,s^{-1}}$ over the test track.

3 Follower Localization

The follower vehicle localization is based on the modified LIO-SAM [14] operating over LiDAR point clouds, where each next pose estimate of the vehicle is output by aligning the current LiDAR scan to the updated map as visualized in Fig. 6. The top-mounted LiDAR is used for the localization.

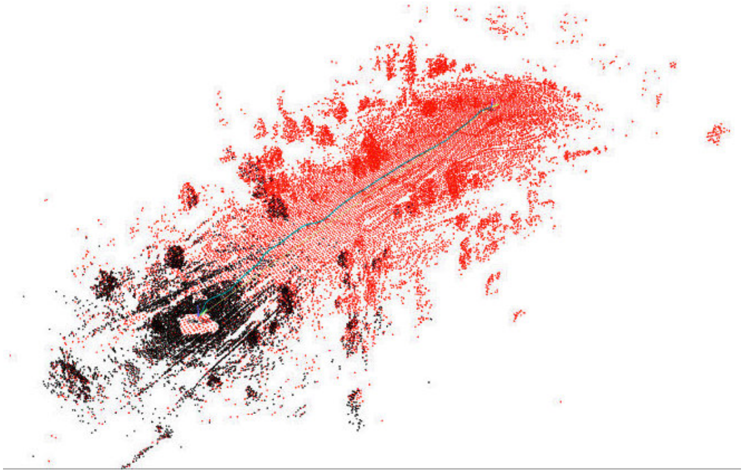


Fig. 6. Building localization map by aligning the current scan (in black) with the previously aligned scans into the point cloud map (in red). The vehicle is traveling from the top right to the bottom left.

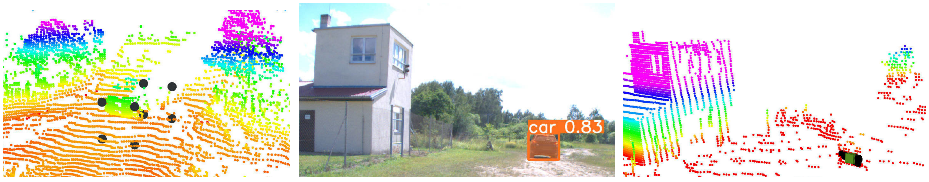
Following adjustments to the original LIO-SAM [14] were made to run the localization adequately onboard the UGV during the field deployments, The initial tilt of the robot (roll and pitch) is estimated by averaging the gravity vectors measured by the IMU for the first 5 s of the deployment. The used IMU is the internal unit of the utilized LiDAR, which does not provide high-quality measurements. Therefore, for the initialization, only the gyroscope readings are used for each initial guess of the rotation, while the initial guess of the translation is taken from the linear motion model. Nevertheless, the provided pose estimates of the follower showed to be sufficient within the leader-follower setup, where the follower primarily follows the relative positions of the leader transformed into the global map of the follower.

4 Proposed Leader-Follower Detection and Tracking

The proposed Leader-Follower system is designed to detect the leader vehicle given the follower's sensor modalities and to track its position within the environment map frame. The sensor modalities to detect the leader are LiDAR point clouds and RGB camera imagery that represent building blocks of the *LiDAR*-based and *Camera-LiDAR*-based detection modules. The detected leader positions, which are observed relative to the robot frame, are transformed into the environment map frame using the localization system. In the map frame, the EKF tracks the detected positions, fusing the leader positions reported by the detection modules. The detected leader path is then passed to the stack, which navigates the robot along the path. The individual building blocks of the Leader-Follower system are described in detail in the rest of the section.

4.1 LiDAR-based and Camera-LiDAR-based Leader Detection

The *LiDAR*-based detection module is based on the CasA network for 3D object detection [20], which is an off-the-github[4] detection system selected based on being the detection component of the best LiDAR-only tracker in the KITTI multi-object tracking benchmark [6]. The detection approach has been opted as a suitable solution for near-field detection with the estimation of the leader orientation. The advantage of the *LiDAR*-based approach is in the estimation of the detected object's relative position without the need for point cloud-to-camera projection or depth estimation, which might suffer from poor camera-LiDAR alignment and poor computational efficiency, respectively. Besides, the detection is reported in the form of oriented 3D bounding boxes, and the *LiDAR*-based approach provides native support for vehicle orientation detection; see an example in Fig. 7a.



(a) LiDAR-based detection (b) Vehicle image segmentation (c) LiDAR points and segmented vehicle

Fig. 7. (a) A LiDAR point cloud (color-coded elevation, red low, purple high) with a bounding box (in black) marking a detected vehicle. (b) Vehicle image segmentation and (c) the LiDAR points (color-coded elevation, red low, purple high) projected onto the segmented vehicle (in black, after filtering) and their centroid (green cube).

The *Camera-LiDAR* detection operates over synchronized RGB camera images and LiDAR point clouds as follows. First, in the images, the leader vehicle is segmented out using YOLOv8 [8] as shown in Fig. 7b. However, image-only detection does not refer to a 3D position without computationally costly depth

[4] https://github.com/hailanyi/CasA

estimation. Hence, image-synchronized point clouds are projected into the images, and the points projected onto the segmented vehicle, visualized in Fig. 7c, are considered to represent the detected vehicle in 3D. Then, the centroid of these points is reported as the vehicle position. Although relatively easy labeling is possible with vehicle segmentation compared to the 3D bounding box of the pure LiDAR-based detection, the camera-LiDAR system is sensitive to camera-LiDAR misalignment.

Due to the misalignment, the LiDAR points projected onto the segmentation are not guaranteed to be close to the detected object. As a result, pixels at the borders of the segmentations may correspond to projected points that are distant from the vehicle. Hence, erosion corresponding to $e^{\text{yolo}} = 25\,\%$ of the bounding box size[5] is applied to remove the potentially spuriously matched pixels at the borders of the segmented vehicle(s). Besides the misalignment, the LiDAR points might go through the leader windows or hollow components that were not segmented out. Since it is assumed that such components are prevalent in the top half of the vehicle, only the bottom half of the segmentation is used.

Additional issues, such as the projection of points corresponding to sparse occluding vegetation that is ignored in the segmentation, are addressed by filtering out vehicle point sets when the centered $q^{\text{yolo}} = 0.8$ quantile of LiDAR-point distance spread is higher than $w^{\text{yolo}} = 3\,\text{m}$ approximately corresponding to the leader width. Besides, the camera-based detection does not provide leader orientation that is beneficial to report when tracking forward-moving vehicles. Therefore, the orientation is approximated as the direction traveled over $\Delta t^{\text{yolo}} = 10\,\text{s}$ window.

4.2 Detection Tracking and Fusion

The detected leader positions are tracked using the EKF tracker coupled to a detection-to-trajectory matching scheme based on [21]. The tracker keeps up to $n^{\text{track}} = 5$ trajectories, such as the one depicted in Fig. 8, representing paths of up to n^{track} vehicles in the follower vicinity. Each trajectory $\mathtt{T} \in \mathcal{T}$ is a time series $\mathtt{T} = \{\boldsymbol{s}_i\}_{i=j}^{k}$ of vehicle states in the form of the normal distribution

$$\boldsymbol{s} = \mathcal{N}(\hat{\boldsymbol{s}}, P), \tag{1}$$

where the tracked state

$$\boldsymbol{s} = (x, y, z, \theta, v, \dot{\theta}) \tag{2}$$

comprises the vehicle position (x, y, z), orientation θ in the horizontal xy plane, forward velocity v, and angular velocity $\dot{\theta}$. Furthermore, $\hat{\boldsymbol{s}}$ is the expected state, and P is the corresponding full covariance matrix. The trajectory time series $\{\boldsymbol{s}_i\}_{i=j}^{k}$ corresponds to the times $\{t_i\}_{i=j}^{k}$, starting at the first vehicle detection associated to the trajectory T at the time t_j and following with the tracker operational frequency $f^{\text{track}} = 1\,\text{Hz}$.

[5] Minimum of bounding box width and height.

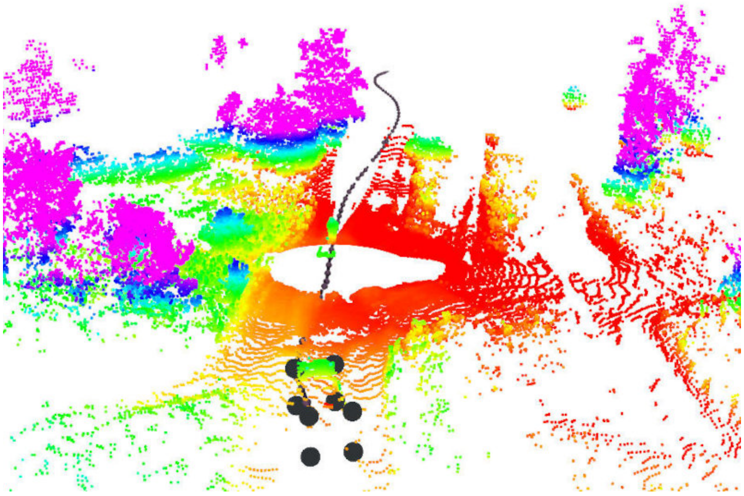


Fig. 8. Leader path tracked using the developed *LiDAR*-based detector and EKF tracker.

At the time t_k, the trajectories are updated w.r.t. the given observation set $Z_k = (Z_k^{\mathrm{LDR}}, Z_k^{\mathrm{CLDR}}) = (\boldsymbol{z}_{k:1}^{\mathrm{LDR}}, \ldots, \boldsymbol{z}_{k:b}^{\mathrm{LDR}}, \boldsymbol{z}_{k:1}^{\mathrm{CLDR}}, \ldots, \boldsymbol{z}_{k:c}^{\mathrm{CLDR}})$ that consists of the b-sized set of *LiDAR*-based detection bounding box centroids $\boldsymbol{z}^{\mathrm{LDR}} = (x^{\mathrm{LDR}}, y^{\mathrm{LDR}}, z^{\mathrm{LDR}}, \theta^{\mathrm{LDR}})$ and the c-sized set of *Camera-LiDAR*-based detection centroids $\boldsymbol{z}^{\mathrm{CLDR}} = (x^{\mathrm{CLDR}}, y^{\mathrm{CLDR}}, z^{\mathrm{CLDR}}, \theta^{\mathrm{CLDR}})$. Since the individual detectors' operational frequencies may differ from f^{track}, the trajectories are updated with an observation modality subset $Z^{\star} \in \{Z^{\mathrm{LDR}}, Z^{\mathrm{CLDR}}\}$ if it is reported within $t_k \pm \frac{\Delta t^{\mathrm{track}}}{2}$, where the timestep $\Delta t^{\mathrm{track}}$ is the inverse of the update frequency $\Delta t^{\mathrm{track}} = \frac{1}{f^{\mathrm{track}}}$. If multiple sets fulfill these conditions, the sets closest to t_k are used.

The observation set Z_k is used to update the trajectories by first the individual observations to the individual trajectories, first, creating new trajectories as needed. Then, the trajectories are updated by the associated observations. The association process and the EKF update are described in detail in the following subsections.

4.3 Detection-Trajectory Association

The individual observations $\boldsymbol{z} \in Z$ are paired to the trajectories $\mathtt{T} \in \mathcal{T}$ w.r.t. the Mahalanobis distance in the 3D space

$$d^{Mxyz}(\boldsymbol{z}^{\star}, \mathtt{T}) = \sqrt{(xyz(\boldsymbol{z}^{\star}) - xyz(\hat{\boldsymbol{s}}))^T xyz(P)^{-1}(xyz(\boldsymbol{z}^{\star}) - xyz(\hat{\boldsymbol{s}}))}, \tag{3}$$

where $xyz(\cdot)$ denotes the subset corresponding to the (x, y, z) dimensions. Since the *LiDAR*-based and *Camera-LiDAR* based detections are independent, the assignments for the two observation types are computed independently. Any particular observation $\boldsymbol{z}^\star \in Z^\star$ is assigned its closest trajectory if it is also the closest observation to the trajectory

$$\boldsymbol{z}^\star \rightarrow \begin{cases} \operatorname{argmin}_{\mathtt{T}\in\mathcal{T}} d^{Mxyz}(\boldsymbol{z}^\star, \mathtt{T}) & \text{if } \boldsymbol{z}^\star = \operatorname{argmin}_{\boldsymbol{z}_j^\star \in Z^\star} d^{Mxyz}(\boldsymbol{z}_j^\star, \mathtt{T}) \wedge d^{Mxyz}(\boldsymbol{z}^\star, \mathtt{T}) < d^{Mxyz}_{\max} \\ \varnothing & \text{otherwise} \end{cases}, \tag{4}$$

where $d^{Mxyz}_{\max} = 1.96$.

Assuming that no observation-trajectory pairs are equidistant, the assignments are considered to be symmetrical, and thus

$$\forall \boldsymbol{z}^\star \in Z^\star, \mathtt{T} \in \mathcal{T} : \boldsymbol{z}^\star \rightarrow \mathtt{T} \Rightarrow \mathtt{T} \rightarrow \boldsymbol{z}^\star. \tag{5}$$

Then, each trajectory is updated with its assigned $\boldsymbol{z}^{\mathrm{LDR}}$ and $\boldsymbol{z}^{\mathrm{CLDR}}$ if at least one of the two is assigned. For any observation that was not assigned to a trajectory, a new trajectory is created if there are fewer trajectories than n^{track}. If trajectories cannot be created for all such observations, the new trajectories are selected arbitrarily.

Finally, trajectories that were not assigned any observation in $l^{\mathrm{track}} = 20$ steps are removed from $\mathcal{T}$. The trajectory considered to be the leader vehicle is initialized at the beginning of the experiment. When the leader trajectory is lost due to l^{track} missed observations, it is rebound to the trajectory of the vehicle closest to the leader.

4.4 EKF Trajectory Update

Each trajectory is considered as an EKF with the transition model

$$\boldsymbol{s}_k = F_k \boldsymbol{s}_{k-1} \tag{6}$$

and the observation model

$$\boldsymbol{z}_k = H_k \boldsymbol{s}_{k-1}. \tag{7}$$

The state transition Jacobian is

$$F_k = \begin{bmatrix} 1\,0\,0\,0 & \sin(\frac{\pi}{2} - \theta_{k-1})\varDelta t^{\mathrm{track}} & 0 \\ 0\,1\,0\,0 & \sin(\theta_{k-1})\varDelta t^{\mathrm{track}} & 0 \\ 0\,0\,1\,0 & 0 & 0 \\ 0\,0\,0\,1 & 0 & \varDelta t^{\mathrm{track}} \\ 0\,0\,0\,0 & 0.99 & 0 \\ 0\,0\,0\,0 & 0 & 0.99 \end{bmatrix}. \tag{8}$$

The observation Jacobian is

$$H_k = \begin{bmatrix} I^{4\times4}\mathbb{I}\left[\boldsymbol{z}^{\mathrm{LDR}} \neq \varnothing\right] & 0^{4\times2} \\ I^{4\times4}\mathbb{I}\left[\boldsymbol{z}^{\mathrm{CLDR}} \neq \varnothing\right] & 0^{4\times2} \end{bmatrix}, \tag{9}$$

where missing observation $\boldsymbol{z}^\star = \varnothing$ is treated as $(0, 0, 0, 0)$ for the purpose of further computations.

The EKF state and covariance prediction is

$$\hat{\boldsymbol{s}}_{k|k-1} = F_k \hat{\boldsymbol{s}}_{k-1|k-1} \tag{10}$$

and

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Iq, \tag{11}$$

respectively, where $q = 1$ scales the process noise. In the EKF update, the gain is

$$G_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R)^{-1} \tag{12}$$

and the observation noise covariance R is assumed to be diagonal $\mathrm{diag}(R) = (\sigma^2(x^{\mathrm{LDR}}), \dots)$, $\sigma^2(x^{\mathrm{LDR}}) = 1000$, and $\sigma^2(x^{\mathrm{LDR}}) = 4000$.

The EKF prediction error is

$$\boldsymbol{e}_k = \boldsymbol{z}_k - H_k \hat{\boldsymbol{s}}_{k|k-1}. \tag{13}$$

Hence, the state and covariance updates are

$$\hat{\boldsymbol{s}}_{k|k} = \hat{\boldsymbol{s}}_{k|k-1} + G_k \boldsymbol{e}_k \tag{14}$$

and

$$P_{k|k} = (I - G_k H) P_{k|k-1}, \tag{15}$$

respectively.

5 Results

The developed leader-follower system has been experimentally verified in a series of field deployments at the Jindřichův Hradec testing site. The site contains several circuits of dirt roads as depicted in Fig. 9 with recognized difficult sections, such as shown in Fig. 10. Specifically, the main challenge for the leader tracker is shown to be bumpy roads on which the leader is frequently out of the follower's sensor sights.

During the field testing in warm June 2024, heat throttling of the main computational processor has been experienced. However, a series of the deployments have been performed since March with increasing max vehicle throttle up to $3\,\mathrm{m\,s^{-1}}$, which corresponds to median velocity approx. $1.5\,\mathrm{m\,s^{-1}}$. The computation is set up so that short, high-velocity periods and zig-zag motion along the path are filtered out of the median velocity and covered distance, respectively. The filtering is applied to provide a covered distance close to the circuit length without such, even small, detours. An overview of the deployments with the achieved median velocities and covered length of the testing course is depicted in Table 1.

The final deployment is considered the one from July 3 that comprised three runs of the LiDAR-based leader detector with the follower maximum throttle set to 1, 2, and $3\,\mathrm{m\,s^{-1}}$, respectively. The LiDAR-based method has been selected due to lower computational requirements, avoiding thermal throttling of the main computational processor.

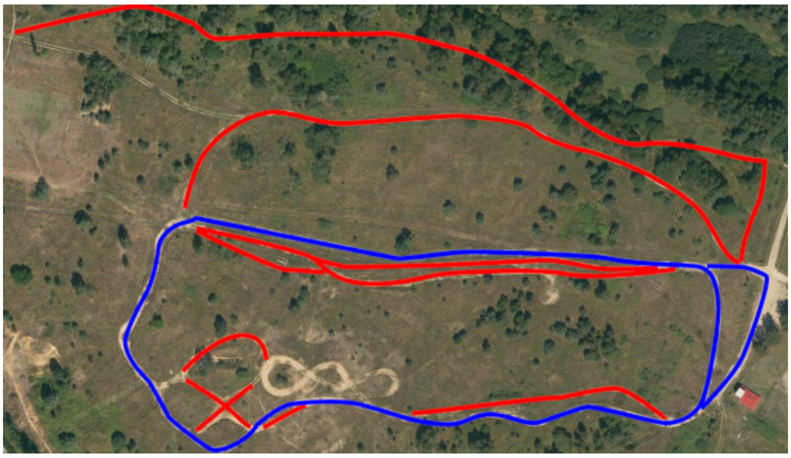
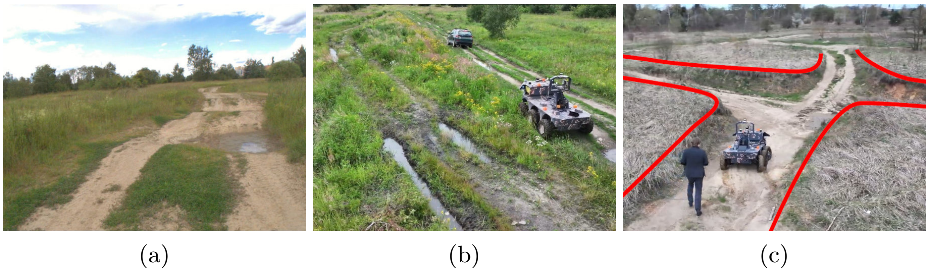


Fig. 9. An overhead view of the field testing site. The selected parts of the site include the main circuit traversed in the autonomous experiments (in blue, about 1 km long) and additional, hard-to-traverse roads that were avoided in the autonomous operation (in red) but used for data collection.



(a) (b) (c)

Fig. 10. The main circuit includes (a) bumpy parts, (b) wet and muddy areas surrounded by the summer vegetation, and (c) x-shaped *sunken crossroad*, the red curves mark the trench edges.

1. The run with $1\,\mathrm{m\,s^{-1}}$ max throttle served to test the system before the faster runs. The UGV traversed the 950 m track in about 40 min, achieving median velocity of $0.49\,\mathrm{m\,s^{-1}}$.
2. In the second run, with the max throttle set to $2\,\mathrm{m\,s^{-1}}$, the circle has been finished in 17 min with a median velocity of $1.01\,\mathrm{m\,s^{-1}}$. The UGV experienced minor heat throttling, leading to missing detections. However, even though it caused the follower to veer off the track on one occasion, the follower was able to recover and finish the circuit without human intervention. Detected leader positions are shown over the map of the environment in Fig. 11.
3. The last run set the max throttle to $3\,\mathrm{m\,s^{-1}}$ to reach the median velocity of $1.52\,\mathrm{m\,s^{-1}}$. However, unlike in the second run, near the end of the experiment,

Table 1. Median velocities achieved during individual experimental runs.

Date	Median velocities [$\mathrm{m\,s^{-1}}$]	Covered distance [m]
March 27, 2024	0.54 0.49	0.08×10^3
April 09, 2024	0.57 0.46	1.11×10^3
April 17, 2024	0.62 0.62 0.61	1.94×10^3
April 24, 2024	0.82 1.04	1.91×10^3
May 24, 2024	1.37 0.89 **1.66**	1.80×10^3
May 28, 2024	1.38 1.05	1.16×10^3
June 17, 2024	0.69	0.12×10^3
July 03, 2024	0.49 1.01 **1.52**	2.50×10^3

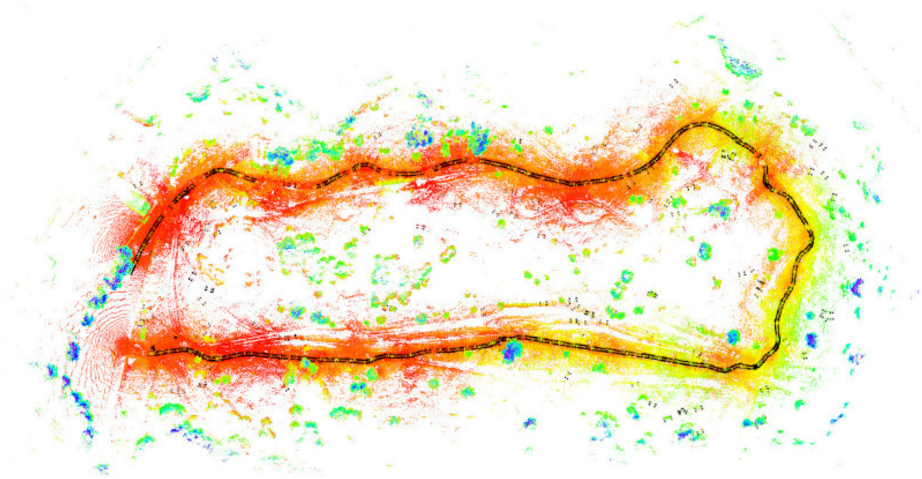


Fig. 11. The detected leader bounding boxes (in black) over the point cloud of the environment (elevation in the jet color scheme) in the second run of the final deployment. In the second half of the clockwise run, heat throttling caused detection outages, which can be seen in the top left of the map.

the heat throttling caused the vehicle to leave the path into a position where the leader tracking could not recovered. Hence, the follower covered only 850 m of the track in about 13 min at the median velocity of 1.52 $\mathrm{m\,s^{-1}}$.

Based on the performed deployments, the current solution showed to be viable and relatively robust for the velocities about 1 $\mathrm{m\,s^{-1}}$.

6 Conclusion

In this paper, we present results on the developed solution of the leader-follower deployment in challenging field conditions with bumpy dirt roads using autonomous follower and crewed leader vehicles. The developed solution is based on the LiDAR-based and Camera-LiDAR-based leader detection accompanied with leader tracking. The follower projects the relative position of the detected leader into its global map, which is being built using SLAM. The projected sequence of the leader's positions is then considered as a path to be followed by the follower; however, the follower may use a local planner with the online updated grid obstacle map to consider its traversability capabilities. The solution has been deployed in a series of trials at the testing site. Although the final deployment has been performed with LiDAR-based only leader detection, the system shown to be feasible and fully autonomous leader-follower has been achieved for about 1 km long test circuit. The planned future work includes the deployment of the Camera-LiDAR-based to improve robustness and increase the speed of the vehicle.

Acknowledgments. The work has been supported by the European Union under the project ROBOPROX - Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590). We would like to thank the 44th Light Motorized Batallion, Army of the Czech Republic, for providing access to the testing site in Jindřichův Hradec, Czechia, our colleagues at the Czech Technical University in Prague and the University of Defence, Czechia, for their comprehensive expertise and advice that contributed toward the project, and the AVT-408 team for their help with the NATO Autonomy Stack, suggestions, and feedback. Namely, we thank Jan Bayer, Bedřich Himmel, Martin Škarytka, Anastázie Rišková, Tulga Ersal, Luboš Hřebecký, and Roman Jarath.

References

1. Bayer, J., Čížek, P., Faigl, J.: Autonomous multi-robot exploration with ground vehicles in darpa subterranean challenge finals. Field Robotics **3**, 266–300 (2023). https://doi.org/10.55417/fr.2023008
2. Belkhouche, F., Belkhouche, B.: Modeling and controlling a robotic convoy using guidance laws strategies. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **35**(4), 813–825 (2005). https://doi.org/10.1109/TSMCB.2005.846646
3. Dani, A.P., Gans, N., Dixon, W.E.: Position-based visual servo control of leader-follower formation using image-based relative pose and relative velocity estimation. In: American Control Conference. pp. 5271–5276 (2009). https://doi.org/10.1109/ACC.2009.5160698
4. Ersal, T., Goodin, C., Sirangelo, D., Balling, O., Haider, M., Gardner, S., Bowen, N., Morales, A., Misko, S., Hughes, I., Verdugo, J.L., Ewing, J., Oommen, T., Wasfy, T., D. Carruth, J.K.: AVT-341 report: Chapter 6 - Autonomy. Tech. rep., NATO Research Task Group AVT-341 (2024)
5. Ersal, T.: Msu-stack, https://github.com/TulgaErsal/MSU-stack, cited on 2024-08-12
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361 (2012). https://doi.org/10.1109/CVPR.2012.6248074

7. Jayakumar, P.: Off-road mobility assessment methods and tools for autonomous military ground systems. Tech. rep., NATO Research Task Group AVT-341 (2023)
8. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (2023), https://github.com/ultralytics/ultralytics
9. Liang, X., Wang, H., Liu, Y.H., Liu, Z., Chen, W.: Leader-following formation control of nonholonomic mobile robots with velocity observers. IEEE/ASME Transactions on Mechatronics **25**(4), 1747–1755 (2020). https://doi.org/10.1109/TMECH.2020.2990991
10. 44th Light Motorized Battalion of General J. Eret, https://44lmopr.army.cz/, cited on 2024-07-26
11. NATO AVT 341 Loyal Wingman full scenario simulation with 4 autonomous ground vehicles, https://www.youtube.com/watch?v=6eklti8ny4g, cited on 2024-07-26
12. Nowakowski, M., Kurylo, J., Braun, J., Berger, G.S., Mendes, J., Lima, J.: Using lidar data as image for ai to recognize objects in the mobile robot operational environment. In: Optimization, Learning Algorithms and Applications. pp. 118–131 (2024). https://doi.org/10.1007/978-3-031-53036-4_9
13. ROS Noetic, https://wiki.ros.org/noetic, cited on 2024-07-26
14. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D.: LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5135–5142 (2020). https://doi.org/10.1109/IROS45743.2020.9341176
15. Stein, P., Spalanzani, A., Santos, V., Laugier, C.: Leader following: A study on classification and selection. Robotics and Autonomous Systems **75**, 79–95 (2016). https://doi.org/10.1016/j.robot.2014.09.028
16. Triest, S., Castro, M.G., Maheshwari, P., Sivaprakasam, M., Wang, W., Scherer, S.: Learning risk-aware costmaps via inverse reinforcement learning for off-road navigation. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 924–930 (2023). https://doi.org/10.1109/ICRA48891.2023.10161268
17. Ubuntu 20.04 LTS, https://ubuntu.com/blog/tag/20-04-lts, cited on 2024-07-26
18. VOP CZ, s.p., https://vop.cz/en/, cited on 2024-07-26
19. Wang, Y., Stanković, M., Smith, A., Matson, E.T.: Leader-follower system in convoys:: An experimental design focusing on computer vision. In: 2021 IEEE Sensors Applications Symposium (SAS). pp. 1–6 (2021). https://doi.org/10.1109/SAS51076.2021.9530146
20. Wu, H., Deng, J., Wen, C., Li, X., Wang, C., Li, J.: CasA: A Cascade attention network for 3-D object detection from LiDAR point clouds. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–11 (2022). https://doi.org/10.1109/TGRS.2022.3203163
21. Wu, H., Han, W., Wen, C., Li, X., Wang, C.: 3D Multi-object tracking in point clouds based on prediction confidence-guided data association. IEEE Transactions on Intelligent Transportation Systems **23**(6), 5668–5677 (2022). https://doi.org/10.1109/TITS.2021.3055616