

A Robust UAV System for Operations in a Constrained Environment

Matěj Petrlik¹, Tomáš Báča¹, Daniel Heřt¹, Matouš Vrba¹, Tomáš Krajník¹, and Martin Saska¹

Abstract—In this letter we present an autonomous system intended for aerial monitoring, inspection and assistance in **Search and Rescue (SAR)** operations within a constrained workspace. The proposed system is designed for deployment in demanding real-world environments with extremely narrow passages only slightly wider than the aerial platform, and with limited visibility due to the absence of illumination and the presence of dust. The focus is on precise localization in an unknown environment, high robustness, safety and fast deployment without any need to install an external infrastructure such as an external computer and localization system. These are the main requirements of the targeted SAR scenarios. The performance of the proposed system was successfully evaluated in the Tunnel Circuit of the DARPA Subterranean Challenge, where the UAV cooperated with ground robots to precisely localize artifacts in a coal mine tunnel system. The challenge was unique due to the intention of the organizers to emulate the unpredictable conditions of a real SAR operation, in which there is no prior knowledge of the obstacles that will be encountered.

Index Terms—Aerial systems: perception and autonomy, search and rescue robots, robotics in hazardous fields.

I. INTRODUCTION

MOBILE robots are being developed to aid human workers in tasks carried out in difficult-to-reach and hazardous environments. Robots can traverse obstructed paths and most importantly, they can minimize the risk of injury by not exposing human workers to dangerous situations. A large research project [1] counts on UAVs to monitor, inspect, diagnose and report on the state of the infrastructure of a city. An example of a use case of this kind is for inspecting the utility infrastructure in road tunnels. Small corridors at the side of the main tunnel that are used to vent smoke and gases during a fire emergency need to be surveyed regularly. This kind of task is risky for human workers, mainly because the primary function of these tunnels cannot be switched off for the inspection time. Any

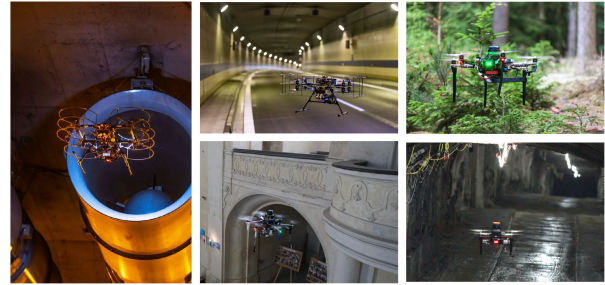


Fig. 1. Deployment of our UAV platforms in constrained environments. From left to right: inspection of ventilation system conditions in road tunnels, monitoring tunnel infrastructure integrity, monitoring dense vegetation, documenting murals not reachable from the ground in churches, and Search and Rescue operations in mines and tunnels.

fire or accident in the main tunnel would therefore have fatal consequences for humans. Hazards can be avoided by using a flying robot to perform the inspection. This solution is especially appealing due to the complicated access to the auxiliary infrastructure and due to its complicated structure including vertical tubes. Other examples of deployments considering the proposed system are for documenting of damaged historical buildings that are dangerous for people to access, and also search and rescue scenarios in difficult-to-access and hazardous environments. This was the main motivation for the research reported in this letter. See Fig. 1 for snapshots of these deployments.

Deploying robots during SAR operations can drastically reduce the time needed to localize victims and thus increase their chance of survival. The area searched for survivors during a fire in a mine can be extended by engaging a robotic team, since human rescuers are not able to continue beyond the first fire. Current localization and mapping algorithms are more capable of providing precise position estimates and a detailed map of the mine, and can therefore provide enhanced situation awareness for the rescue mission plan. For these reasons, the **Defense Advance Research Projects Agency (DARPA)** hosts the Subterranean Challenge robotic competition (SubT), aimed at pushing research on SAR robotics forward. The goal of the Tunnel Circuit, the first SubT, was to send robots into mining tunnels to find survival-specific objects and to find survivors, referred to together as artifacts. The DARPA artifacts consisted of a total of 60 objects, e.g., effigy of a survivor, backpack, fire extinguisher, electric drill, and cell phone. The robotic team was employed to detect the artifacts, and to report their exact position to a base station located at the entrance to the tunnel. One point was awarded for each detected artifact reported with precision of 5 m or less.

The design of the proposed system is motivated by SAR requirements and challenges, together with the assumption that

Manuscript received September 10, 2019; accepted January 14, 2020. Date of publication February 3, 2020; date of current version February 17, 2020. This letter was recommended for publication by Associate Editor F. Ruggiero and Editor J. Roberts upon evaluation of the reviewers' comments. This work was supported by CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics", and by CTU Grant SGS17/187/OHK3/3T/13. The work of Tomáš Krajník was supported by CSF Grant 17-27006Y. The work of Matěj Petrlik and Martin Saska was supported by CSF Grant 17-16900Y. (Corresponding author: Matej Petrlik.)

The authors are with the Faculty of Electrical Engineering, Czech Technical University in Prague, 166 36 Prague 6, Czechia (e-mail: petrmat@fel.cvut.cz; tomas.baca@fel.cvut.cz; hertdani@fel.cvut.cz; vrbamato@fel.cvut.cz; tomas.krajnik@agents.fel.cvut.cz; saska@labe.felk.cvut.cz).

This article has supplementary downloadable material available at <https://ieeexplore.ieee.org>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.2970980

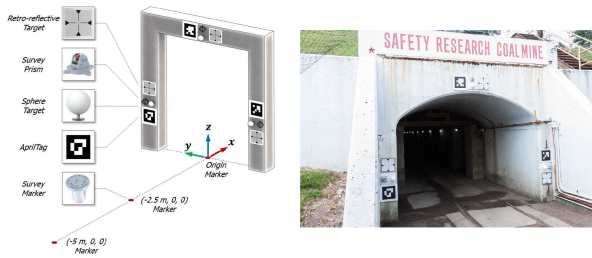


Fig. 2. The fiducial markers at the entrance gate define the origin of the SubT frame of reference in which the artifacts must be localized.

nobody may enter the mine before the robots. This reaches far beyond the research currently being carried out in robotics. The possibility to deploy robotic systems in real mines and in conditions absolutely the same as in a real mining disaster has provided a unique opportunity to develop a realistic robotic system. In comparison with standard laboratory experiments, the evaluation of robotic algorithms was much harsher in SubT. The number of trials each team could make was limited, and the teams were not allowed to choose when their run would begin. These demands matched the requirements of real SAR scenarios, where the system has to be ready to use shortly after the arrival of the rescuers, and no repeated trials are possible.

In this letter focused on UAVs, we describe a system that is able to precisely self-localize the robots in the tunnel map, which is gradually built during the run. The origin of the map frame must be aligned to the entrance gate marked by fiducial markers, as shown in Fig. 2, for the purpose of evaluating the localization accuracy of the reported artifacts.

The system has to be robust to unexpected obstacles in an unknown environment and in difficult conditions (absolute darkness, presence of dust and smoke). Various artifacts need to be identified in real time by onboard sensors and their position needs to be estimated precisely. The data (the type and the position of the detected artifacts) need to be delivered to the tunnel entrance, where a human operator is located.

A. Related Work

An overview of the current state of SAR robotics is presented in [2]. Until very recently the deployment of UAVs in SAR scenarios has been restricted to the use of teleoperated drones for scanning the sites of natural disasters (e.g., cities after an earthquake [3], flooded areas [4], forest fires [5]) from a high altitude, and therefore far from obstacles and able to use GNSS. Systems intended for working in cluttered, GNSS-denied environments are mostly focused on tunnel [6] and chimney [7] inspection applications. Obviously, none of these systems was focused on the requirements of the SAR scenarios under discussion here, with their demanding and completely unknown environment, the need for a rapid response, and a high level of complexity. In the field of SAR, only some preliminary studies of searching in collapsed buildings can be considered as relevant to the proposed system. Platforms with adaptive morphology [8] are capable of flying through narrow gaps in the ruins of buildings.

In GNSS-denied locations, UAVs are often self-localized using an image stream from one or more onboard camera. Visual simultaneous localization and mapping (VSLAM) [9], and visual inertial odometry (VIO) [10] are methods that have received special attention. However, cameras require favorable

lighting conditions to function optimally [11], due to the limited dynamic range of the sensor.

For insufficiently illuminated environments, methods based on laser scan matching, e.g., Generalized ICP [12], NDT [13] and MRSMs [14] are typically more suitable. A thorough experimental evaluation of scan registration algorithms for mapping purposes is presented in [15]. In [16], IMU and LIDAR measurements are used to estimate the position by matching consecutive scans while simultaneously building a map, and [17] combines laser and visual odometry (VO) to develop more reliable hypotheses for various parts of the flight.

Autonomous navigation of UAVs deployed in forest rescue missions is made possible by applying clustering to the laser scans and finding tree-shaped features, as described in [18]. For more general unknown environments, reinforcement learning can be used [19]. Exploration of an unknown space with the goal of maximizing the information gain about objects of interest is achieved by an RRT-based planner in [20]. Free space observations in 3D are often incomplete, yielding a large number of unexplored regions (frontiers), and this limits the exploration performance. The performance can be improved by considering only occupied space to determine the frontiers, using stochastic differential expansion of particles [21].

Detection of artifacts or object of interests typically relies on RGB cameras. Recently, traditional approaches for general object detection in a RGB camera images, have mostly been surpassed by new methods based on deep learning, such as YOLO [22], Faster-RCNN [23] and SSD [24] Convolutional Neural Networks (CNNs). Deep learning detectors offer various trade-offs between speed, precision and robustness, based on the structure of the specific neural network. For our purposes, we have selected YOLO [22], which offers good performance, while being less computationally demanding than the other models. It is therefore able to run online on the onboard computer of the UAV.

The positions of detected artifacts need to be transferred to the command post operator. A combination of WiFi and ZigBee wireless technologies in a cluster-tree network topology was introduced by [25]. When reliable communication cannot be achieved, each member of a UAV team can adopt a specific strategy to be able to relay the data from the explored sites [26], or to employ a controller that ensures the availability of communication resources [27].

B. Contribution

The main contribution of this letter is the complete UAV system capable of mapping and navigating in complex constrained environments such as tunnels, mines, corridors and hallways. The key element of our solution is the robustness to errors in localization that are bound to happen in such challenging conditions. The proposed control scheme sacrifices the tracking precision of aggressive controllers such as the SO(3) [28] in favor of more fault-tolerant model predictive control approach [29], while it preserves the non-linear capabilities of SO(3). Compared to our previous approach [29], the proposed controller handles state and input constraints, it is capable of control far outside of the linearization point (the state of hover) and generates feedforward action using the known UAV mass.

The performance and the robustness of the system has been demonstrated in challenging conditions of the DARPA SubT. Moreover, applications such as documentation of historical

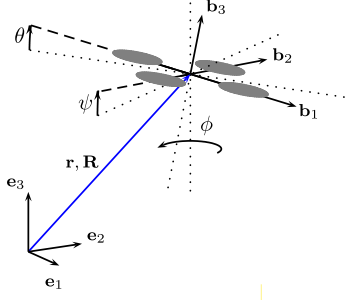


Fig. 3. The world frame $w = \{e_1, e_2, e_3\}$, in which the position and orientation of the UAV is expressed by translation $\mathbf{r} = [x, y, z]^T$ and rotation $\mathbf{R}(\phi, \theta, \psi)$ to the body frame $b = \{b_1, b_2, b_3\}$.

objects [30] would not be possible without a reliable system, as a failure would result in damage to the building inventory.

In this letter, we claim that **contrary to established understanding, a 2D localization method is not only sufficient** for navigating in 3D man-made tunnel environments, but it achieves similar or higher robustness than 3D state-of-the-art visual-based methods in conditions typical for SAR. The 2D localization method profits from the similarity of the horizontal cross-section of the tunnel among different heights. Our hypothesis is supported by two independently-evaluated experiments. At the first one, performed at the DARPA STIX event, the **2D LIDAR-based method outperformed the 3D VO method in robustness, precision and map quality** characterised by (19). The second one, performed at the DARPA SubT event confirmed the robustness of our approach, which allowed successful navigation through the previously unvisited man-made tunnel environments, where state-of-the-art approaches and even dedicated localization hardware failed (Section III-A).

We share here our architecture and the approach taken to tackle the Tunnel Circuit of the DARPA SubT along with problems that were encountered, lessons learned, ideas and experience gained, which we hope will be beneficial for the robotic community, especially for researchers pursuing the topic of SAR aerial robotics. For a broader overview of the whole robotic team deployed at DARPA SubT, see [31].

II. AUTONOMOUS SYSTEM

The autonomous system is composed from individual modules shown in Fig. 4. A description of the architecture of the system follows.

A. Position Estimation

The crucial aspect of autonomous flying in a constrained environment is to precisely estimate the UAV state vector

$$\mathbf{x} = (\mathbf{r}, \dot{\mathbf{r}}, \ddot{\mathbf{r}}, \mathbf{R}, \dot{\mathbf{R}})^T, \quad (1)$$

where $\mathbf{r} = [x, y, z]^T$ is the UAV position, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$ are the linear velocity and the acceleration vectors, respectively, $\mathbf{R}(\phi, \theta, \psi)$ is the orientation in the world coordinate frame, and $\dot{\mathbf{R}}, \ddot{\mathbf{R}}$ are the angular velocities (see Fig. 3). The state variables $\ddot{\mathbf{r}}, \mathbf{R}, \dot{\mathbf{R}}$ are explicitly available from the proprioceptive sensors contained in the Inertial Measurement Unit (IMU).

The first derivative of orientation $\dot{\mathbf{R}}$ is measured by the gyros of the IMU, and the second derivative of position $\ddot{\mathbf{r}}$ is measured directly by the accelerometers. With known acceleration, the

orientation is obtained from the model

$$m\ddot{\mathbf{r}} = f\mathbf{R}\mathbf{e}_3 + m\mathbf{g}\mathbf{e}_3. \quad (2)$$

The UAV mass is denoted by $m \in \mathbb{R}$. The magnitude of the gravitational pull is denoted by $g \in \mathbb{R}$, and $f \in \mathbb{R}$ represents the magnitude of the total thrust force generated by the propellers. Only \mathbf{r} and $\dot{\mathbf{r}}$ are left to be estimated in order to have the complete state representation of the UAV.

As the components of $\mathbf{r} = [x, y, z]^T$ are independent, the estimate can be decoupled for each axis, which decreases size of the state-space description. The z -component is obtained from rangefinder-barometer fusion in a linear Kalman filter scheme, and the filters for the x and y axes have an identical model, so only x -axis estimation will be discussed for brevity. The state vector for the x -axis is defined as $\mathbf{x} = (x, \dot{x}, \ddot{x}, \ddot{x}^{(u)}, \ddot{x}^{(d)}, \theta)^T$, where $\ddot{x}^{(u)}$ is the acceleration caused by the control, $\ddot{x}^{(d)}$ is the acceleration from external disturbance forces (wind, imperfect propellers, etc.), and θ is the rotation around the y -axis of the world coordinate frame. The state vector is estimated by constructing a state observer of a discrete linear system $\mathbf{x}_{[t+1]} = \mathbf{A}\mathbf{x}_{[t]} + \mathbf{B}\mathbf{u}_{[t]}$, with the system matrix $\mathbf{A} \in \mathbb{R}^{6 \times 6}$ and the input matrix $\mathbf{B} \in \mathbb{R}^{6 \times 1}$:

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & p_3 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & p_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ p_2 \end{bmatrix}, \quad (3)$$

where p_1 is the proportional term relating acceleration and attitude of a linearized UAV model operating around the hover state. First-order transfer parameters from the desired attitude angle to the actual attitude angle are denoted by p_2, p_3 . The linearized UAV model is sufficient for the estimation due to the absence of aggressive maneuvers that would excite the system beyond the operating point.

The estimation loop runs at 100 Hz driven by the control input $\mathbf{u}_t = \theta^{(u)}$, i.e., the desired tilt angle. Since the desired tilt angle is not always achieved exactly, the orientation from (2) is used to correct the actual tilt state variable. Direct measurements of acceleration are degraded by propeller-induced vibrations, so it is not used to correct the acceleration state variable. The velocity and the position are obtained by integration realized by the model, and as such they are prone to drift, so a position or velocity correction is necessary to stabilize the system. Position corrections from Hector SLAM [32] are used to localize the UAV in the reference frame with its origin at the takeoff position and with the x -axis pointing in the direction in which the UAV is facing.

Hector SLAM is a motion estimation algorithm developed specifically for **urban search and rescue (USAR)** scenarios. The laser scans are first transformed into a local stabilized coordinate system using the UAV orientation measured by IMU. After filtering out the laser scan endpoints with the distance from the horizontal plane passing through the UAV body frame larger than a threshold value, the points are aligned into the online-estimated occupancy grid map. Bilinear filtering is employed for calculating the occupancy probabilities and the derivatives, which achieves sub-grid cell accuracy. The size of grid cells was set to 4 cm. The algorithm lacks SLAM backend pose graph optimization, as the accuracy is satisfactory even without loop closure, and this saves computational resources.

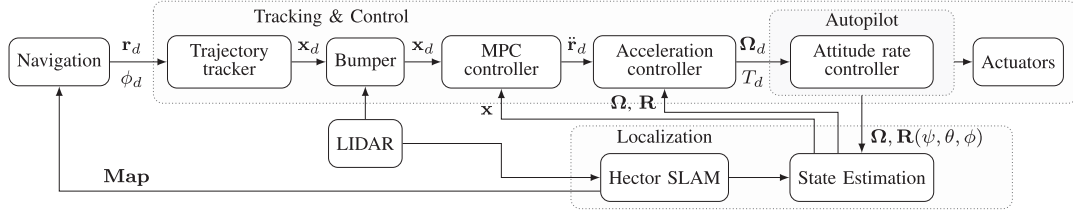


Fig. 4. The interconnection of individual blocks of the proposed system architecture from high-level planning to UAV motor actuators.

B. Automatic Control

The hierarchical approach presented here utilizes the internal attitude rate feedback loop in the Pixhawk flight controller with the PX4 control stack. The attitude rate reference Ω_d and the desired thrust t_d for Pixhawk is generated by a non-linear controller based on the SO(3) approach in [28]. We augment SO(3) and replace its feedback terms with a specially designed input \ddot{x}_d , which is obtained from a linear Model Predictive Control scheme. Such combination creates control scheme, which is very robust to jumps in both the estimated state and the control reference thanks to the MPC controller. The non-linear SO(3) feedback controller still provides excellent acceleration tracking in any given state of the UAV. A system of body and world disturbance estimators is used to separate the external forces from internal biases of the flight controller.

1) *UAV Model*: The UAV model relies on the notation of the coordinate frames from Fig. 3, and is described as

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v}, & m\dot{\mathbf{v}} &= f\mathbf{R}\mathbf{e}_3 + m\mathbf{g}\mathbf{e}_3, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\Omega}}, & \mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} &= \mathbf{M}, \end{aligned} \quad (4)$$

where $\mathbf{r} = [x, y, z]^T$ is the position of the UAV in w , $\mathbf{R}(\phi, \theta, \psi)$ is the orientation of the UAV in w , $\boldsymbol{\Omega} \in \mathbb{R}^3$ is the angular velocity in b and the *hat map* $\hat{\cdot} : \mathbb{R}^3 \rightarrow SO(3)$ is defined by the condition $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$. A gravitational force with magnitude $g \in \mathbb{R}$ acts on the vehicle together with the thrust force created by the propellers, the magnitude of which is denoted by $f \in \mathbb{R}$. $m \in \mathbb{R}$ stands for the UAV mass. The angular velocity of the UAV in the body frame is $\boldsymbol{\Omega} = [\omega_1, \omega_2, \omega_3]^T \in \mathbb{R}^3$ and $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ is the inertia matrix. The total moment exerted by the propellers onto the UAV is $\mathbf{M} = [M_1, M_2, M_3]^T$.

2) *Acceleration Tracking SO(3) Controller*: First, we derive the desired force acting on the UAV as

$$\mathbf{f}_d = (m(\ddot{\mathbf{r}}_D - g\mathbf{e}_3) - \mathbf{d}_w - \mathbf{d}_b)\mathbf{R}\mathbf{e}_3, \quad (5)$$

where m [kg] is the UAV mass, $\ddot{\mathbf{x}}$ [ms^{-2}] is the desired acceleration, g [ms^{-2}] is the magnitude of the gravitational acceleration, and \mathbf{d}_w and \mathbf{d}_b are the world and body disturbance force terms. The disturbance terms are integrals of the apparent force acting on the UAV in the body frame b and in the world frame w . They are obtained as:

$$\mathbf{d}_w = \int_0^t ki_w \mathbf{e}_w d\tau, \quad \mathbf{d}_b = \int_0^t ki_b \mathbf{R}(\tau)^T \mathbf{e}_w d\tau, \quad (6)$$

where ki_w and ki_b are the body integral gains and the world integral gains, respectively, and $\mathbf{e}_w = \mathbf{r} - \mathbf{r}_d$ is the control error in the world frame. The force \mathbf{f}_d serves for calculating the desired attitude

$$\mathbf{R}_d = [\mathbf{b}_{1d}, \mathbf{b}_{2d}, \mathbf{b}_{3d}], \quad (7)$$

which is composed of vectors

$$\mathbf{b}_{3d} = \|\mathbf{f}\|, \quad \mathbf{b}_{2d} = \frac{\mathbf{b}_{3d} \times \mathbf{R}(\phi_d)_0}{\|\mathbf{b}_{3d} \times \mathbf{R}(\phi_d)_0\|}, \quad \mathbf{b}_{1d} = \mathbf{b}_{2d} \times \mathbf{b}_{3d}, \quad (8)$$

where $\mathbf{R}(\phi_d)_0$ is the first column of the rotational matrix created from the desired yaw ϕ_d and $\theta = 0, \psi = 0$ Euler angles. We then calculate the orientation error as

$$\mathbf{e}_R = \frac{1}{2} (\mathbf{R}_d^T \mathbf{R} - \mathbf{R}^T \mathbf{R}_d). \quad (9)$$

Finally, the desired angular rate is obtained as

$$\boldsymbol{\Omega}_d = -k_R \mathbf{e}_R - k_\Omega \mathbf{e}_\Omega,$$

where $\mathbf{e}_\Omega = \mathbf{R}^T \mathbf{R}_d \boldsymbol{\Omega}_d$ is a feedforward term representing the attitude rate error. The control action consists of the desired thrust $t_d = \mathbf{f} \cdot \mathbf{b}_3 \in [0, 1]$ and the desired attitude rate $\boldsymbol{\Omega}_d$.

3) *Robust Feedback Through Acceleration Command*: The linear Model Predictive Control (MPC) is a robust feedback method for a system with a known model. We utilize MPC for controlling the constrained translational dynamics of the UAV by generating the desired acceleration command \ddot{x}_d . MPC uses an LTI model with n states and k inputs:

$$\mathbf{x}_{[t+1]} = \mathbf{A}\mathbf{x}_{[t]} + \mathbf{B}\mathbf{u}_{[t]}, \quad \mathbf{y}_{[t]} = \mathbf{C}\mathbf{x}_{[t]} + \mathbf{D}\mathbf{u}_{[t]}, \quad (10)$$

where $\mathbf{r} \in \mathbb{R}^n$ denotes the state vector and $\mathbf{u} \in \mathbb{R}^k$ denotes the input vector. Matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times k}$ are the main system matrix and the input matrix, respectively. In contrast with the traditional full description of the LTI system, we assume $\mathbf{C} = \mathbf{I}$, and $\mathbf{D} = \mathbf{0}$. Therefore $\mathbf{y}_{[t]} = \mathbf{x}_{[t]}$ holds. The matrices in equation (10) used in MPC differ from the ones on observer model (3) as it does not contain variables specific for state estimation. An MPC control error is defined along a reference trajectory of length i as $\mathbf{e}_{[i]} = \mathbf{x}_{[i]} - \hat{\mathbf{x}}_{[i]}$, where $\hat{\mathbf{x}}$ is the trajectory reference. MPC is formulated as a quadratic programming problem as

$$\min_{\mathbf{u}_{[t]}, \mathbf{x}_{[t]}} V(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \sum_{i=1}^{m-1} (\mathbf{e}_{[i]}^T \mathbf{Q} \mathbf{e}_{[i]} + \mathbf{u}_{[i]}^T \mathbf{P} \mathbf{u}_{[i]}) \quad (11)$$

$$\text{s.t. } \mathbf{x}_{[t+1]} = \mathbf{A}\mathbf{x}_{[t]} + \mathbf{B}\mathbf{u}_{[t]}, \quad \forall t \in \{0, \dots, m-1\} \quad (12)$$

$$\mathbf{x}_{[t]} \leq \mathbf{x}_{\text{max}[t]}, \quad \forall t \in \{1, \dots, m\} \quad (13)$$

$$\mathbf{x}_{[t]} \geq \mathbf{x}_{\text{min}[t]}, \quad \forall t \in \{1, \dots, m\} \quad (14)$$

where the quadratic cost function (11) penalizes the control error and the input action over a horizon $m \in \mathbb{Z}^+$ in length. $\mathbf{Q} \succcurlyeq 0$ and $\mathbf{P} \succcurlyeq 0$ are state error and input penalization matrices. Constraint (12) forces the states to follow model (10), while constraints (13) and (14) bound the states of the dynamical system.

A state-space representation of the translational dynamics in the world frame is described by state vector \mathbf{x} as

$$\begin{aligned} \mathbf{x}[1] &= \mathbf{r}[1], & \mathbf{x}[4] &= \mathbf{r}[2], & \mathbf{x}[7] &= \mathbf{r}[3], & \mathbf{x}[10] &= \phi, \\ \mathbf{x}[2] &= \dot{\mathbf{r}}[1], & \mathbf{x}[5] &= \dot{\mathbf{r}}[2], & \mathbf{x}[8] &= \dot{\mathbf{r}}[3], & \mathbf{x}[11] &= \dot{\phi}, \\ \mathbf{x}[3] &= \ddot{\mathbf{r}}[1], & \mathbf{x}[6] &= \ddot{\mathbf{r}}[2], & \mathbf{x}[9] &= \ddot{\mathbf{r}}[3], & \mathbf{x}[12] &= \ddot{\phi}. \end{aligned} \quad (15)$$

The four controllable states are then expressed as a differentially flat discrete system with matrices \mathbf{A} and \mathbf{B} defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_s & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_s \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_s(\Delta t) \\ \mathbf{B}_s(\Delta t) \\ \mathbf{B}_s(\Delta t) \\ \mathbf{B}_s(\Delta t) \end{bmatrix}, \quad (16)$$

where sub-system matrices \mathbf{A}_s and \mathbf{B}_s are defined as

$$\mathbf{A}_s = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_s = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \\ 1 \end{bmatrix}, \quad (17)$$

With this system, the acceleration reference $\ddot{\mathbf{r}}_d$ for the tracking controller presented in II-B2 is extracted directly from $\mathbf{u}_{[0]}$, i.e., the first control input of the MPC.

C. Trajectory Generation and Safety

The input to the control pipeline is a path from a grid-based planner (Sec. II-D). Firstly, the path is naively sampled with desired speed and acceleration. A *trajectory tracker* is then used to obtain a feasible reference. The *MPC tracker* [33] was used to post-process the trajectory in real time. The tracker creates a reference, which obeys the dynamics up to snap and jerk. The reference originates from the current desired state of the system and obeys the input and state constraints. The tracker then supplies the current reference state to the *MPC controller* at 100 Hz. Since obstacles may suddenly appear in the LIDAR sensor data, we employed a reactive virtual bumper to avoid them. The sensor data is first filtered by the temporal median filter. When an obstacle is detected in the filtered sensor data within a safety radius around the UAV, the *trajectory tracker* is suspended until a dedicated repulsion maneuver takes the UAV away from the danger of a collision.

D. Exploration and Navigation

The 2D occupancy grid produced by Hector SLAM is the basis for navigation and exploration. It distinguishes three states of cells: free, obstacle and unexplored. The grid is first downsampled to a lower resolution (25 cm), to reduce computation times, obstacles are inflated, and then a distance transform is applied over the grid, to obtain the distance to the closest obstacle for each cell of the grid. The modified A* algorithm is used to plan trajectories in the grid at 2 Hz rate using the heuristic:

$$h(\mathbf{x}) = \text{dist}(\mathbf{x}, \mathbf{g}) + o(\mathbf{x})p, \quad (18)$$

where \mathbf{x} is the current evaluated point in the grid, \mathbf{g} is the goal, $\text{dist}(\mathbf{x}, \mathbf{g})$ is a Euclidean distance, $o(\mathbf{x})$ is the distance to the closest obstacle, obtained from the distance transform, and p is a tunable parameter, which controls how much the planning algorithm avoids obstacles. In the first (exploratory) part of the flight, unknown cells in the grid are considered as free, in order to promote exploration of unknown areas.

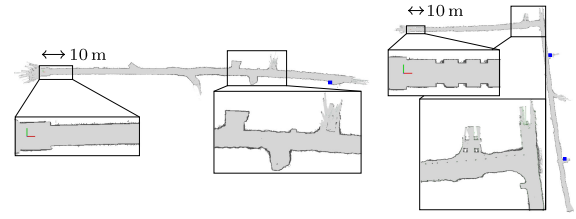


Fig. 5. The occupancy grid map estimated by Hector SLAM during flights in Experimental Tunnel (left) and Safety Research Tunnel (right). Blue squares depict the estimated positions of artifacts.



Fig. 6. Artifacts detected during the flights in the Tunnel Circuit. From left to right: drill from day 2 in the Experimental Tunnel, cell phone and survivor from day 4 in the Safety Research Tunnel.

Initially, the goal and the flight height (1 m in the case of DARPA SubT) is set by the operator and the UAV explores in the set direction, according to the planning algorithm. If the *Return strategy* is chosen for the mission (see Section II-F), the UAV returns to its initial position, by setting the starting point as a new goal for the planning algorithm after half of the battery power has been exhausted. Otherwise, the UAV continues exploring until it drains all available battery power. When returning, all unexplored cells in the grid are declared as obstacles, to force the UAV to go via a known path.

The exploration is not entirely uninformed clearing of frontiers, but it is biased by the direction of the initially set goal. This prevents situations when e.g., a mine has more entrances, the robot enters through one of them and starts clearing frontiers in the other entrances, instead of continuing further into the tunnels.

E. Artifact Detection and Localization

The artifacts are detected online in images from the onboard camera using the YOLO Convolutional Neural Network (CNN) [22], which is a state-of-the-art object detector CNN. The CNN runs on the onboard computer of the UAV at 1 Hz rate. It was trained on a dataset, which was obtained in an abandoned mine [31] in order to closely emulate the environment of the competition.

The output of the YOLO CNN is a set of object candidates in the input image as shown in Fig. 6. Each candidate consists of an estimate of the bounding rectangle of the object in the image, its class, and the confidence.

1) *Artifact Position Estimation*: It is assumed that artifacts are placed either on the ground or on the wall of the tunnel. This assumption is used to estimate the position of an artifact candidate from its bounding rectangle in the image and from the limited sensory equipment onboard the UAV.

The undistorted center of the bounding rectangle is projected to a 3D ray using a pinhole camera model. The distances of the ground and the wall of the tunnel are estimated using the onboard

2D laser scanner and rangefinder. Intersections of the 3D ray with the ground and wall are calculated. The closest intersection that lies in the direction in which the camera is facing is assumed to be the approximate position of the detected artifact.

2) *Artifact Candidate Filtering*: Artifact candidates from different images are grouped on the basis of the Euclidean distance of their estimated 3D positions. The position of each candidate group is filtered using a Kalman Filter. The class and the confidence of each group is obtained as the average of the corresponding candidates. When the confidence of a group and the number of candidates in the group exceed a threshold, the group is considered to be a confirmed detection of an artifact. Confirmed detections are transferred to the command post, as described in the next section. In the competition, the system was tuned for high sensitivity at the cost of an increased number of false positives, because these were manually filtered by the operator.

F. Wireless Communication

The detected artifacts need to be transferred to the stationary command post outside of the tunnel. The composition of SAR robotic rescue team affects the suitable strategy. Two independent communication strategies were implemented:

1) *Return Strategy*: The UAV explores forward into the tunnel using one half of its battery life, and then returns to upload the detections into the command post CouchDB [34] database over WiFi. This strategy does not rely on other robots, hence the UAV can be the first agent sent into the tunnel. Moreover, long-distance communication is also not needed, which eliminates the risk of mission failure due to communication problems. On the other hand, if the UAV fails to return, the operator will not obtain any data.

2) *Mesh Modules*: The UAV carries a small radio module (EU - 868 MHz, US - 915 MHz) that forms a mesh network with modules on other deployed robots. In heterogeneous robot teams the ground robots also carry more battery-powered modules, which are dropped when the received signal strength indication (RSSI) drops below a threshold value. If the UAV does not have to return to the command post, it has the full battery time available for exploration and can therefore reach much deeper into the tunnel.

III. EXPERIMENTAL EVALUATION

The UAV system was evaluated during two robotic events: The SubT Integration Exercise (STIX) and the SubT Tunnel Circuit. Each event presented different challenges in the environment. A high resolution video attachment to this work is available at: <http://mrs.felk.cvut.cz/icra2020darpa>

A. Discussion of Results

During the competition flights, the UAV explored 340 m of the mine tunnels and found 3 artifacts (Fig. 6) out of the 10 artifacts discovered by all our robots. A summary of each day is presented in Table I. Parameters of the system were adjusted based on the first trials to reflect the environment better. This resulted in the performance of the system gradually improving, as is evident from Table I.

The competing teams encountered the drawbacks of visual localization methods. Intel RealSense T265 is considered a state-of-the-art out-of-the-box hardware localization solution

TABLE I
SUMMARY OF DISTANCE FLOWN AND ARTIFACTS FOUND ON EACH COMPETITION DAY. UAV DATA FOR THE THIRD DAY IS MISSING IN THE TABLE, AS OUR TEAM ALREADY HAD 5 POINTS FROM THIS TUNNEL FROM THE PREVIOUS DAY, SO AFTER TWO SHORT FLIGHTS NEAR THE ENTRANCE TO THE TUNNEL TO VERIFY THE CHANGES TO THE ALGORITHM PARAMETERS WE DECIDED NOT TO SEND THE UAV FURTHER INTO THE TUNNEL

	Day 1	Day 2	Day 3	Day 4
Tunnel	Safe.	Exp.	Exp.	Safe.
Distance reached by UAV [m]	115	120	-	220
Flight time [mm:ss]	4:09	10:05	-	12:05
Mean Map Entropy [nat]	-2.22	-2.14	-	-2.32
Artifacts found by UAV	0	1	-	2
Total artifacts found	1	5	5	5

for robotics, and indeed preliminary results on legged robots were promising [35]. However the poor performance in characteristic SAR conditions (dark, dust) degraded even more by propeller-induced vibrations proved the shortcomings of 3D visual localization in these environments. The team Coordinated Robotics relied on the otherwise precise T265 during SubT, but the insufficient robustness caused a localization failure before reaching any artifact. Multiple teams relied on a 3D LIDAR, but only one of them reported to achieve a similar distance of ~ 200 m asation about the surroundings than a 2D LIDAR, it is unsuitable for small UAVs with limited payload deployed in a spatially-constrained environment. To the best of our knowledge, we were the only team to rely on a 2D LIDAR and 2D localization that, thanks to its robustness, helped us to surpass the performance of other teams.

B. UAV Platform

Despite the platform-agnostic software architecture, we present the specifications of the platform used for the Tunnel Circuit. A quadrotor was purposefully designed for the DARPA SubT and SAR projects. The low-level attitude rate control and IMU measurements are supplied by Pixhawk flight controller. Higher-level control, localization, navigation, exploration, and detection algorithms are executed on the onboard Intel NUC computer. The main source of exteroceptive data is RPLiDAR A3, a planar LiDAR rotating at 13 Hz. The height above the terrain is measured by Garmin Lidar Lite v3 attached to the bottom of the UAV. A single front-facing mvBlueFox-MLC200 w global shutter camera running at 2 Hz is used for the artifact detections and for reference frame alignment. The camera field of view is illuminated by a 20 W LED light. A small radio frequency transceiver (RFM69HW) is used for communication with other robots and with the mission command post. The UAV automatically initiates the exploration within 30 s of pressing the power-on button.

C. Lessons Learned and Future Work

The most challenging problems that we had to overcome during the Tunnel Circuit and STIX were:

- 1) The constrained flight corridor, with narrow passages that are too tight for the UAV to pass through
- 2) The large amount of dust blown around by the propellers prevents the use of visual localization

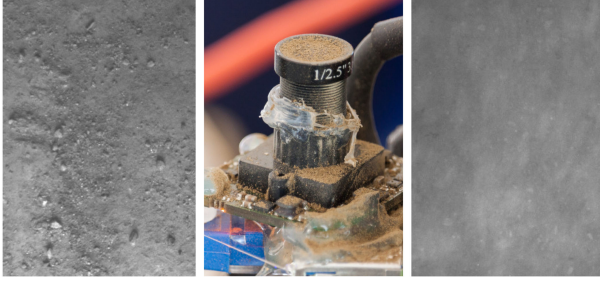


Fig. 7. The clear image from the optical flow camera (left) was degraded by the layer of dust on the lens (middle). After approx. 5 min of flight the camera stream became unusable (right).

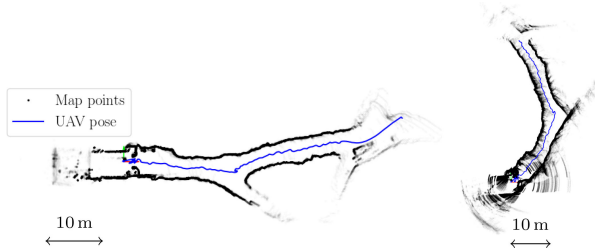


Fig. 8. The point clouds aggregated from the LIDAR scans when the UAV was localized by Hector SLAM (left) and by optical flow (right) during the same flight in the STIX event. The Hector SLAM point cloud is significantly more consistent (sharp).

- 3) The featureless walls in parts of the tunnels have a negative impact on LIDAR localization
- 4) Aggressive controllers with precise reference tracking destabilize the unreliably localized UAV

The first of these issues led to the design of a smaller quadrotor with F450 frame instead of the F550 hexarotor. The F550 was able to pass through 1.8 m wide entrance gate reliably, but struggled with the narrowest passage that was 1.2 m wide. The minimal width of feasible passage depends on the repulsion parameters of the virtual bumper.

Second, we did not anticipate the amount of dust present in the STIX tunnels. It was expected that the floor would be wet from dripping water, so mud and pools of water were a greater concern for us. However, the tunnel was much drier than we assumed, and this caused an immense amount of dust to be blown around by the propellers. Visual sensing was degraded to the point that stabilizing the UAV with a custom optical flow algorithm [36] was not feasible after a layer of dust had settled on the lens (Fig. 7).

We decided to move from camera-based localization to LIDAR-based localization as the laser technology is less prone to dust [37]. In addition, it was possible to remove the downward-pointing LED-matrix which served as a light source for the optical flow algorithm in adverse lighting conditions. To fit into the reliability requirement of the competition, we chose the Hector SLAM [32] LIDAR localization method, which achieved consistent performance in real world experiments [38], [39]. Despite the lack of ground truth data from the STIX event, we provide a comparison between LIDAR scans aggregated into a global point cloud with optical flow and Hector SLAM localization in Fig. 8. We calculated the map mean entropy

(MME) [40] as

$$H(\mathbf{M}) = \frac{1}{|\mathbf{M}|} \sum_{k=1}^{|\mathbf{M}|} \frac{1}{2} \ln |2\pi e \Sigma(q_k)|, \quad (19)$$

where $\Sigma(q_k)$ is the sample covariance of mapped points in a local radius $r = 0.3$ m around the map point q_k . It has been shown in [40] that the sharpness of the map represented by the MME correlates with the error of the trajectory when the ground truth pose of the UAV is available, and as such it can be used to approximate the position estimate quality in the absence of ground truth. The map obtained with Hector SLAM achieved much better -2.04 nat MME than optical flow with -1.52 nat MME, which validated our decision to use LIDAR localization instead of VO for the Tunnel Circuit.

The unreliability of optical flow localization also led to the development of a fault-tolerant MPC controller, which does not destabilize the UAV during occasional localization glitches (see II-B). Nevertheless, the trade-off for higher robustness of the MPC controller is less precise tracking of the control reference and higher control error in comparison with the previously-used SO(3) controller [28]. The traditional control approaches, such as PID, or the SO(3) [28], minimize the control errors regardless of dynamic constraints. That is sufficient when the state estimate and the control reference obey the constraints, and in such cases, the control can be precise and aggressive. However, this can result in producing too aggressive maneuvers in case of large control errors, which can occur with unreliable localization systems, such as in this challenge. The proposed control approach maintains a stable and smooth flight even with arbitrary estimator jumps. In practice, we experienced estimator jumps of several meters while flying within the tunnels. As an added benefit, it is also robust to infeasible reference, which can theoretically consist of a step reference in position. While this could be achieved using a traditional MPC controller, our approach produces proper control signals even outside of the point of linearization (the hover state). Although the UAV should avoid reaching such states, a traditional MPC controller cannot guarantee that the UAV will not reach it due to external disturbance.

The walls at the beginning of the Experimental Tunnel of the Tunnel Circuit were smooth, without any protruding elements (see Fig. 5). As a result, the covariance of the Hector SLAM position estimate grew in the direction of the tunnel. The localization became unreliable, and this in turn caused incorrect map alignment. During the flights in the Safety Research Tunnel, the UAV was much better localized thanks to the rectangular pillars around the tunnel wall.

Both tunnels of the Tunnel Circuit were less dusty than the tunnels at STIX, so optical flow would have been a viable option especially in Experimental Tunnel, where the Hector SLAM performance suffered from the absence of features on the smooth tunnel walls. The position estimation framework presented in Section II allows multiple state measurement sources to be fused with dynamic confidence. We plan to combine the advantages of the LIDAR-and-camera-based localization techniques to obtain a reliable state estimate in all situations. Furthermore, we intend to modify the LIDAR localization by assigning higher weights to laser scan endpoints with normals orthogonal to the tunnel direction.

We have exploited the assumption that the horizontal cross-section of the tunnel does not change significantly among different heights, which allowed the use of 2D LIDAR and 2D SLAM. However, the next round of the DARPA SubT will be an Urban Circuit, which will be more vertically diverse, with stairwells and ceilings of varying heights. Hence, the 2D localization strategy will not be feasible, and it will be necessary to equip the UAV with a 3D sensor and 3D localization method. Moreover, both the Urban Circuit and the Cave Circuit are expected to consist of large areas, in which the range of the current laser scanner may not be sufficient.

IV. CONCLUSION

We have developed a complex UAV system for operations in hard-to-reach hazardous places such as emergency vents in road tunnels, dam pipelines, mining tunnels, and chimneys. Missions in such environments require high reliability, and in the case of SAR they require also a minimal preparation time. The proposed system is operational and ready to start the mission in 30 s after pressing the power-on button. In this letter, we have presented that 2D localization is sufficient for navigating 3D man-made tunnel environments. The robustness of the system was proven in the Tunnel Circuit of the Subterranean Challenge organized by DARPA, where we achieved 1st place among non-funded teams and 3rd place in total.

REFERENCES

- [1] I. Robotics, "Robotic and autonomous systems for resilient infrastructure," *UK-RAS White Papers*© UK-RAS, 2017.
- [2] J. Delmerico *et al.*, "The current state and future outlook of rescue robotics," *J. Field Robot.*, vol. 36, no. 7, pp. 1171–1191, 2019.
- [3] S. Verykokou *et al.*, "3D reconstruction of disaster scenes for urban search and rescue," *Multimedia Tools Appl.*, vol. 77 no. 8, pp. 9691–9717, Apr. 2018.
- [4] M. Rahnemounfar, R. Murphy, M. V. Miquel, D. Dobbs, and A. Adams, "Flooded area detection from uav images based on densely connected recurrent neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 1788–1791.
- [5] C. Yuan, Z. Liu, and Y. Zhang, "UAV-based forest fire detection and tracking using image processing techniques," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2015, pp. 639–643.
- [6] T. Özarslan *et al.*, "Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1740–1747, Jul. 2017.
- [7] M. Nieuwenhuisen, J. Quenzel, M. Beul, D. Droschel, S. Houben, and S. Behnke, "ChimneySpector: Autonomous MAV-based indoor chimney inspection employing 3D laser localization and textured surface reconstruction," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2017, pp. 278–285.
- [8] D. Falanga, K. Kleber, S. Mintchev, D. Floreano, and D. Scaramuzza, "The foldable drone: A morphing quadrotor that can squeeze and fly," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 209–216, Apr. 2019.
- [9] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [10] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2502–2509.
- [11] H. Alismail, M. Kaess, B. Browning, and S. Lucey, "Direct visual odometry in low light using binary descriptors," *IEEE Robot. and Automat. Lett.*, vol. 2, no. 2, pp. 444–451, Apr. 2017.
- [12] A. Segal, D. Hähnel, and S. Thrun, "Generalized-icp," in *Proc. Robot.: Sci. Syst.*, vol. 2, no. 4, p. 435, 2009.
- [13] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.
- [14] J. Stückler and S. Behnke, "Multi-Resolution Surfel Maps for Efficient Dense 3D Modeling and Tracking," *J. Visual Commun. Image Representation*, vol. 25, no. 1, pp. 137–147, Jan. 2014.
- [15] J. Razlaw, D. Droschel, D. Holz, and S. Behnke, "Evaluation of registration methods for sparse 3D laser scans," in *Proc. IEEE Eur. Conf. Mobile Robots*, 2015, pp. 1–7.
- [16] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, "Lidar-inertial integration for uav localization and mapping in complex environments," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2016, pp. 649–656.
- [17] T. Tomic *et al.*, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. & Automat. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.
- [18] J. Q. Cui, S. Lai, X. Dong, P. Liu, B. M. Chen, and T. H. Lee, "Autonomous navigation of uav in forest," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2014, pp. 726–733.
- [19] H. X. Pham, H. M. La, D. Feil-Seifer, and L. Van Nguyen, "Reinforcement learning for autonomous uav navigation using function approximation," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2018, pp. 1–6.
- [20] T. Dang, C. Papachristos, and K. Alexis, "Autonomous exploration and simultaneous object search using aerial robots," in *Proc. IEEE Aerosp. Conf.*, 2018, pp. 1–7.
- [21] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3D exploration with a micro-aerial vehicle," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 9–15.
- [22] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 6517–6525.
- [23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [24] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Proc. Eur. Conf. Comput. Vision*, pp. 21–37, 2016.
- [25] P. Tao and L. Xiaoyang, "Hybrid wireless communication system using zigbee and wifi technology in the coalmine tunnels," in *Proc. IEEE Third Int. Conf. Measuring Technol. Mechatronics Automat.*, 2011, pp. 340–343.
- [26] K. Cesare, R. Skeelee, Soo-Hyun Yoo, Yawei Zhang, and G. Hollinger, "Multi-uav exploration with limited communication and battery," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 2230–2235.
- [27] J. Fink, A. Ribeiro, and V. Kumar, "Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams," in *Proc. IEEE*, vol. 100, no. 1, pp. 164–178, Jan. 2012.
- [28] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. IEEE Conf. Decis. Control*, 2010, pp. 5420–5425.
- [29] T. Baca, G. Loianno, and M. Saska, "Embedded model predictive control of unmanned micro aerial vehicles," in *Proc. Int. Conf. Methods Models Automat. Robot.*, 2016, pp. 992–997.
- [30] M. Saska, V. Krátký, V. Spurný, and T. Báča, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," in *Proc. Int. Conf. Emerg. Technologies Factory Automat.*, 2017, pp. 1–8.
- [31] T. Rouček *et al.*, "Darpa subterranean challenge: Multi-robotic exploration of underground environments," in *Proc. MESAS*, 2019.
- [32] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3D motion estimation," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2011, pp. 155–160.
- [33] T. Baca, D. Hert, G. Loianno, M. Saska, and V. Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles," in *Proc. Int. Conf. Intell. Robots Syst.*, 2018, pp. 6753–6760.
- [34] J. C. Anderson, J. Lehnardt, and N. Slater, *CouchDB: The Definitive Guide: Time to Relax*. Philadelphia, PA, USA: O'Reilly Media, Inc., 2010.
- [35] J. Bayer and J. Faigl, "On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265," in *Eur. Conf. Mobile Robots*, 2019, pp. 1–6.
- [36] V. Walter, T. Novák, and M. Saska, "Self-localization of unmanned aerial vehicles using onboard sensors, with focus on optical flow from camera image," in *Proc. MESAS*, 2017, pp. 106–132.
- [37] X. Zhu, P. Church, and M. Labrie, "Lidar for obstacle detection during helicopter landing," in *Proc. SPIE DSS*, vol. 6950, 2008, pp. 194–201.
- [38] J. M. Santos, D. Portugal, and R. P. Rocha, "An evaluation of 2d slam techniques available in robot operating system," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2013, pp. 1–6.
- [39] J. M. Santos, M. S. Couceiro, D. Portugal, and R. P. Rocha, "A sensor fusion layer to cope with reduced visibility in slam," *J. Intell. & Robot. Syst.*, vol. 80, no. 34, pp. 401–422, Dec. 2015.
- [40] D. Droschel, J. Stückler, and S. Behnke, "Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 5221–5226.