2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Control-Barrier-Aided Teleoperation with Visual-Inertial SLAM for Safe MAV Navigation in Complex Environments

Siqi Zhou[1,4], Sotiris Papatheodorou[2,3,4], Stefan Leutenegger[2,3,4], Angela P. Schoellig[1,4]

***Abstract*— In this paper, we consider a Micro Aerial Vehicle (MAV) system teleoperated by a non-expert and introduce a perceptive safety filter that leverages Control Barrier Functions (CBFs) in conjunction with Visual-Inertial Simultaneous Localization and Mapping (VI-SLAM) and dense 3D occupancy mapping to guarantee safe navigation in complex and unstructured environments. Our system relies solely on onboard IMU measurements, stereo infrared images, and depth images and autonomously corrects teleoperated inputs when they are deemed unsafe. We define a point in 3D space as unsafe if it satisfies either of two conditions: *(i)* it is occupied by an obstacle, or *(ii)* it remains unmapped. At each time step, an occupancy map of the environment is updated by the VI-SLAM by fusing the onboard measurements, and a CBF is constructed to parameterize the (un)safe region in the 3D space. Given the CBF and state feedback from the VI-SLAM module, a safety filter computes a certified reference that best matches the teleoperation input while satisfying the safety constraint encoded by the CBF. In contrast to existing perception-based safe control frameworks, we directly close the perception-action loop and demonstrate the full capability of safe control in combination with real-time VI-SLAM *without* any external infrastructure or prior knowledge of the environment. We verify the efficacy of the perceptive safety filter in real-time MAV experiments using exclusively onboard sensing and computation and show that the teleoperated MAV is able to safely navigate through unknown environments despite arbitrary inputs sent by the teleoperator.**

I. Introduction

MAVs have proven to be very versatile platforms capable of aiding in a variety of tasks such as large-scale inspection, search and rescue, construction site monitoring or photography and cinematography. While there is ongoing research on performing these kinds of tasks in an autonomous manner, direct teleoperation of the MAV by a human is at times desirable or necessary. It is thus important to ensure safe MAV navigation not only during autonomous flight but also under direct teleoperation.

Safe autonomous flight and teleoperation differ in where the MAV reference path or trajectory originates from but have otherwise similar requirements. In both cases, a map of the MAV's environment is required so that regions where it is safe to fly can be determined. Creating a useful map is dependent upon the accurate localization of the MAV. A safety-aware controller is needed to ensure no reference path causes the MAV to collide by taking the map and current MAV state estimate into account. Finally, in order to allow deployment of the system in a wide range of environments, it is desirable for localization, mapping, and control to be performed onboard the MAV using its sensors.

In this paper, we propose a system combining onboard localization, mapping and safe control, which allows collision-free teleoperation of an MAV in complex environments. To the best of our knowledge, this is the first system that

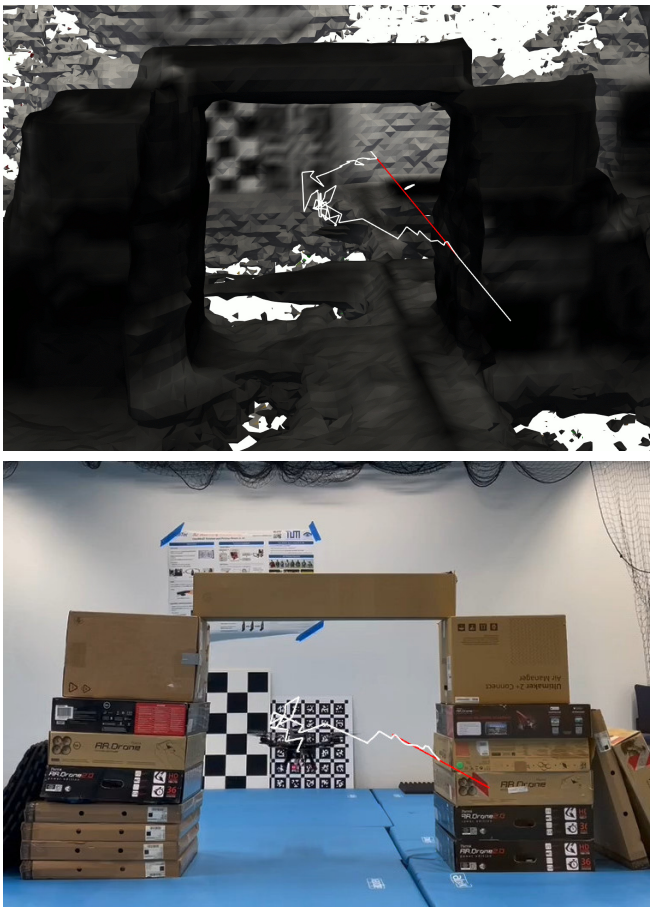

Fig. 1. [Top] The operator sends a series of position references in a straight line (red). The safety filter modifies the reference (white) taking the onboard volumetric map into account. [Bottom] Photograph of the real-world MAV with the commanded (red) and filtered (white) position references overlaid.

[1]Learning Systems and Robotics Lab, School of Computation, Information and Technology, Technical University of Munich. E-mail addresses: {siqi.zhou, angela.schoellig}@tum.de

[2]Smart Robotics Lab, School of Computation, Information and Technology, Technical University of Munich. E-mail addresses: {sotiris.papatheodorou, stefan.leutenegger}@tum.de

[3]Smart Robotics Lab, Department of Computing, Imperial College London. E-mail addresses: {s.papatheodorou18, s.leutenegger}@ic.ac.uk

[4]Munich Institute of Robotics and Machine Intelligence (MIRMI).

This work was supported by the Technical University of Munich, MIRMI, and the EU Horizon project DigiForest.

combines a Control Barrier Function (CBF)-based safety filter with onboard, online localization and dense volumetric mapping so that no prior infrastructure or knowledge of the environment is required. In summary, we propose the following contributions:

- The generation of a Control Barrier Function (CBF) from an online dense volumetric map constructed in real-time using onboard sensors, directly closing the loop between mapping and control.
- The integration of VI-SLAM, dense 3D occupancy mapping, and safe control within an MAV, establishing a self-contained system that operates without the need for external infrastructure.
- A series of experiments conducted in both simulated environments and real-world scenarios to showcase the efficacy and robustness of our proposed system.

II. Related Work

We present a brief overview of the most relevant works in VI-SLAM, volumetric mapping, and safe control.

A. Visual-Inertial Odometry and SLAM

Visual-Inertial Odometry (VIO) is the process of estimating a robot's state given camera images and Inertial Measurement Unit (IMU) readings. The IMU measurements are especially important in the case of MAVs which can exhibit fast movement that may cause a visual-only odometry system to lose track of the MAV's state. Visual-Inertial SLAM expands on VIO by also performing loop closure upon place recognition and therefore keeping consistent estimates of trajectory and map.

One of the first approaches to combine visual with inertial measurements was MSCKF [1], a filtering-based approach for adding geometric constraints from visual observations into an inertial navigation algorithm. There have been several extensions to this method since (e.g., OpenVINS [2]) and other notable filter-based approaches (e.g., ROVIO [3]). It is extended in maplab [4] to perform localization against large-scale sparse maps.

Non-linear optimization based approaches such as BASALT [5], VINS-Fusion [6] and ORB-SLAM3 [7] are also popular due to their improved accuracy. In Kimera [8], a sparse VI-SLAM system is combined with dense mapping using meshes. OKVIS [9] is a lightweight optimization-based VIO method that is extended in OKVIS2 [10] to include loop closures and semantic segmentation for the filtering of dynamic objects. Due to its state-of-the-art accuracy and runtime performance, we use it in our approach.

B. Volumetric Mapping

Volumetric mapping refers to the creation of a dense 3D map of an environment which can then be used for SLAM or planning. In the seminal work of KinectFusion [11], the mapped space is represented as a discretized Truncated Signed Distance Field (TSDF) stored in a fixed-size grid. Newer approaches have improved the scalability of the map by using voxel hashing [12] or octrees [13] instead of a grid. TSDF-based maps don't explicitly represent *free* space and thus cannot be used directly for safe navigation. Voxblox [14] solves this issue by incrementally creating a Euclidean Signed Distance Field (ESDF) from the TSDF map to allow safe onboard motion planning; however, it does not directly close the loop between mapping and control.

Another approach is to perform occupancy mapping, which entails storing the occupancy probability of each point in the mapped space. This has the benefit that *occupied*, *free* and *unknown* regions of the environment can be explicitly represented, allowing safe path planning. OctoMap [15] is a popular occupancy mapping framework that has been the basis of other methods such as UFOMap [16], which extends it to explicitly store *unknown* space. FIESTA [17] incrementally creates an ESDF map suitable for MAV motion planning from an occupancy map. Another octree-based occupancy mapping method is supereight [13] and its extension [18], which introduces propagation of occupancy values through all levels of the octree for efficient path planning.

C. Safe Control

Control theory has been widely used to provide desired safety guarantees for dynamical systems. In recent years, there has been an increasing number of work on learning-based methods to further address the challenge of guaranteeing the safety of the system in the presence of uncertainties [19]. Examples include but are not limited to Gaussian Process Model Predictive Control (GP-MPC) [20], GP-based robust control [21], neural adaptive control [22], and learning-based Control Barrier certification [23]. While classical control and more advanced learning-based control methods have proven effective, most of them rely on having sufficiently accurate state feedback and a clear understanding of the environment [19]. In practical applications, robot systems are subject to noisy state estimation and incomplete knowledge about their surrounding environments. In these cases, they are required to distill safety constraints in real-time from high-dimensional sensory inputs and comply with safety constraints inferred from perception.

There exist works targeted to bridge the gap between perception and safe control. For instance, a robust CBF formulation was introduced in [24] to account for perception errors, a differentiable barrier network was proposed in [25] to incorporate a CBF quadratic program (QP) in an end-to-end learning pipeline, and a perceptive safe locomotion approach was presented in [26] to infer steppable planes and safe actions based exteroceptive sensors. In this work, we similarly aim to bridge the perception-action gap with a particular emphasis on demonstrating the capabilities of a safe control system that incorporates VI-SLAM and dense 3D mapping in the closed loop for safe teleoperated MAV navigation in unstructured and cluttered environments.

III. Problem Statement

We consider an MAV that tracks reference positions produced by a human operator. Our goal is to ensure the MAV safely navigates cluttered environments and avoids collisions

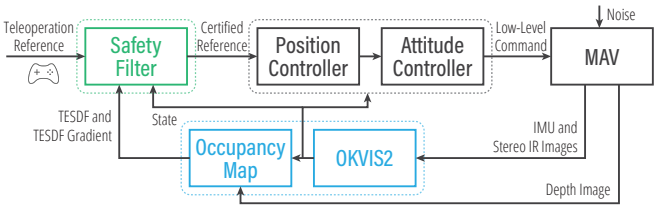

Fig. 2. A block diagram of the MAV system. The system consists of three main modules: *(i)* a baseline controller for reference tracking, *(ii)* a perception module (blue) that keeps an updated dense 3D map of the environment and provides estimates of the MAV state in real-time, and *(iii)* a safety filter (green) that adjusts the teleoperated inputs based on perceptive information when they are deemed unsafe. Our goal is to design a perceptive safety filter to guarantee the safety of the teleoperated MAV system in complex and unstructured environments.

even when the operator reference positions would cause it to collide. Additionally, the MAV should be prevented from entering unmapped regions. To achieve these, we use the MAV's onboard sensors to localize within the environment, create a map of safely navigable regions, and use a safety filter to modify the operator reference positions as required to avoid constraint violations. We assume that the MAV is equipped with an IMU, a stereo camera and a depth sensor.

A. Environment Model

The static environment is modelled as a bounded volume $\mathbb{V} \subset \mathbb{R}^3$ with each point $\mathbf{x} \in \mathbb{V}$ having an associated occupancy probability $P_o(\mathbf{x})$. We define the *free* , *unknown* and *occupied* sets as $\mathbb{V}_{\text{free}} = \{\mathbf{x} \in \mathbb{V},\ P_o(\mathbf{x}) < 0.5\}$, $\mathbb{V}_{\text{unkn}} = \{\mathbf{x} \in \mathbb{V},\ P_o(\mathbf{x}) = 0.5\}$ and $\mathbb{V}_{\text{occup}} = \{\mathbf{x} \in \mathbb{V},\ P_o(\mathbf{x}) > 0.5\}$.

B. MAV Control System

The MAV control system computes desired attitude commands based on position references. We assume that the closed-loop MAV system can be locally approximated by a discrete-time linear model:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \approx \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k, \tag{1}$$

where $k \in \mathbb{Z}_{\geq 0}$ denotes discrete-time index, $\mathbf{x} \in \mathbb{V} \subset \mathbb{R}^3$ is the position of the MAV system, $\mathbf{u} \in \mathbb{U} \subset \mathbb{R}^3$ is the position reference input, $\mathbf{w}_k$ is random noise, $\mathbf{f} : \mathbb{V} \times \mathbb{U} \mapsto \mathbb{V}$ is a Lipschitz continuous function, and $(\mathbf{A}, \mathbf{B})$ are matrices of consistent dimensions.

In this work, we assume that the noise in the system is bounded as follows: $||\mathbf{w}_k|| \leq \varepsilon$ with $\varepsilon \in \mathbb{R}_{\geq 0}$, where $||\cdot||$ denotes the Euclidean norm. When the control system has perfect tracking, the system matrices $\mathbf{A}_k$ and $\mathbf{B}_k$ would be a zero matrix and an identity matrix, respectively. When perfect tracking is not achieved, one may learn the system matrices $(\mathbf{A}_k, \mathbf{B}_k)$ based on data collected online to account for the dynamics of the underlying closed-loop system.

IV. Methodology

Our approach combines VI-SLAM, dense 3D mapping, and a safety filter, all running online and onboard the MAV. The OKVIS2 [10] VI-SLAM system receives grayscale stereo image pairs and IMU measurements and produces MAV state estimates. Depth images and corresponding pose estimates are used by supereight 2 [18] to create a dense 3D occupancy map. The control system of the MAV consists of a position controller and an attitude controller, which together serve as a tracking controller that converts desired position signals to lower-level commands. The lower-level commands are then realized by the MAV autopilot. The safety filter layer utilizes the map generated by the perception system and minimally adjusts the reference sent to the controller to ensure collision avoidance with obstacles and remaining within mapped regions. A block diagram of the overall system architecture can be seen in Fig. 2.

A. Visual-Inertial SLAM

We use the OKVIS2 [10], a state-of-the-art, sparse, optimization-based VI-SLAM system for MAV state estimation. OKVIS2 consists of a frontend and a real-time estimator that processes multiple images and IMU messages synchronously whenever a new frame arrives. The front end performs keypoint matching, stereo triangulation and place recognition which triggers loop closures if successful. The real-time estimator optimizes the current factor graph and marginalizes old observations. OKVIS2 also performs optimization of the full factor graph on loop closures asynchronously and then synchronizes the result with the real-time factor graph. We further utilize the real-time IMU propagation capabilities of OKVIS2 to obtain the most up-to-date state estimates at IMU rate, numerically integrating from the newest optimized state.

B. Dense Mapping

We use a modified version of the multi-resolution mapping pipeline from [18] to create a dense volumetric occupancy map of the environment. Depth measurements are integrated in a probabilistic manner to account for sensor noise, and *free* space is explicitly mapped to allow safe MAV navigation. It is important to note that for collision-free MAV movement, only *free* regions of the map must be considered safe since *unknown* regions may contain yet unmapped obstacles.

In order to use the occupancy map in a CBF, a differentiable field must be extracted from it. The occupancy field is discontinuous at the boundaries between *free* and *unknown* space, as shown in Fig. 3 [Left], making it unsuitable for use with a CBF. A space representation more suitable for use with a CBF is the Truncated Euclidean Signed Distance Function (TESDF) $h(\mathbf{x}) : \mathbb{V} \to \mathbb{R}$ which is differentiable over $\mathbb{V}$ and defined as

$$h(\mathbf{x}) = \begin{cases} \min\left(d(\mathbf{x},\ \partial\mathbb{V}_{\text{free}}),\ h_b\right), & \text{if } \mathbf{x} \in \mathbb{V}_{\text{free}}, \\ -\min\left(d(\mathbf{x},\ \partial\mathbb{V}_{\text{free}}),\ h_b\right), & \text{otherwise}, \end{cases} \tag{2}$$

where $h_b \in \mathbb{R}^+$ is the truncation bound, ∂V denotes the boundary of set $\mathbb{V}$ and $d(\mathbf{x}, \mathbb{V})$ the Euclidean distance of point $\mathbf{x}$ from set $\mathbb{V}$. This results in the TESDF having positive values in the interior of the safe set (i.e., *free* space), negative values outside the safe set (i.e., *occupied* or *unknown* space), and 0 at the safe set boundary. The truncation bound h_b is used to limit the volume affecting the TESDF of a given

point, allowing more efficient computation of TESDF values in a volumetric map.

We extend the mapping framework to allow querying $h(\mathbf{x})$ and $\nabla h(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{V}$. The queried TESDF values and gradients are computed online from the latest occupancy map data to ensure safe navigation. A TESDF slice through the map and the corresponding occupancy slice are shown in Fig. 3.

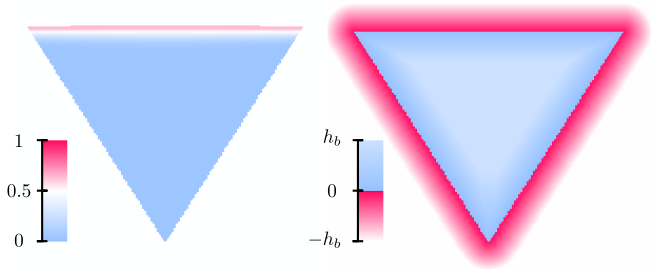

Fig. 3. [Left] Occupancy field slice after integration of a single depth image with *free*, *occupied*, and *unknown* space shown in shades of blue, red and white, respectively. [Right] Corresponding TESDF slice with positive (safe) and negative (unsafe) values shown in shades of blue and red, respectively. Notice the occupancy field discontinuity between *free* and *unknown* space compared to the smooth transition of the TESDF.

C. Safety Filter Design

We employ a discrete-time CBF formulation in the design of our safety filter. To facilitate our discussion, we first introduce the case where $\mathbf{w}_k = \mathbf{0}$ and then extend the framework to account for noise in the system.

A discrete-time CFB safety filter can be formulated as an optimization problem [27]:

$$\min_{\mathbf{u}} \quad ||\mathbf{u} - \mathbf{u}_{\text{teleop}}||^2 \tag{3a}$$

$$\text{subject to} \quad \Delta h(\mathbf{x}_k, \mathbf{u}) \geq \alpha\, h(\mathbf{x}_k), \tag{3b}$$

where $\Delta h(\mathbf{x}_k, \mathbf{u}) = h\big(f(\mathbf{x}_k, \mathbf{u})\big) - h(\mathbf{x}_k)$ is the change in the value of the TESDF after an input $\mathbf{u}$ is applied, and $\mathbf{u}_{\text{teleop}}$ is the teleoperation input. The term α is defined as follows:

$$\alpha = \begin{cases} -p_1, & \text{if } h(\mathbf{x}_k) \geq 0, \\ p_2, & \text{otherwise}, \end{cases} \tag{4}$$

where p_1 and p_2 are positive constants. The safety filter in (3) aims to find an input $\mathbf{u}$ that is close to the teleoperation input $\mathbf{u}_{\text{teleop}}$ while guaranteeing the safety constraint represented by the CBF condition is satisfied.

Intuitively, the CBF condition (3b) is a scalar condition that guarantees the positive invariance of a set (i.e., if the system starts in the set, it will remain in the set). When the system is in the safe set, the condition has two implications: *(i)* at the safety boundary, the change in h is lower bounded by 0, which implies that the system either remains on the boundary or moves towards the interior of the safe set, and *(ii)* in the interior of the safe set, the change in h is lower bounded by a negative number which quantifies how fast the system is allowed to approach the safety boundary. When the system is outside of the safe set, the change in h is required to be positive, which is intended to move the system closer to the safe set and ultimately return to the safe set.

The CBF constraint in the optimization problem in (3) is generally nonlinear in the decision variable $\mathbf{u}$ and is not trivial to solve. To simplify the problem, we use a first-order approximation of the CBF constraint, which allows us to formulate the safety filter optimization problem as a quadratic program (QP) that can be solved efficiently online. The first-order approximation of the CBF constraint can be written as follows:

$$\mathbf{C}\,\mathbf{u} \geq c_1, \tag{5}$$

where $\mathbf{C}{=}\nabla^\intercal h(\mathbf{x}_k)\,\mathbf{B}_k$ and $c_1{=}\nabla^\intercal h(\mathbf{x}_k)\,(\mathbf{I}-\mathbf{A}_k)\,\mathbf{x}_k + \alpha h(\mathbf{x}_k)$.

When the system dynamics are noisy, one can robustly account for the noise in the design of the CBF condition to guarantee safety in the presence of uncertainties [24]. We introduce a robust CBF condition in our safety filter design. By introducing the term $\mathbf{w}$ to Δh and following the same linearization procedure, we obtain the following robustified condition:

$$\mathbf{C}\,\mathbf{u} \geq c_1 + c_2, \tag{6}$$

where $c_2 = ||\nabla h(\mathbf{x}_k)||\varepsilon$. This condition increases the lower bound on Δh by a positive number that is proportional to the level of noise in the system. Intuitively, the term c_2 characterizes the worst-case uncertainty in Δh due to system noise. The overall CBF safety filter is

$$\min_{\mathbf{u}} \quad ||\mathbf{u} - \mathbf{u}_{\text{teleop}}||^2 \tag{7a}$$

$$\text{subject to} \quad \mathbf{C}\,\mathbf{u} \geq c_1 + c_2, \tag{7b}$$

which has a standard QP form.

To account for the spatial extent of the MAV, one can generally offset the TESDF such that $h \geq \underline{d}$ is guaranteed, where $\underline{d}$ characterizes the dimension of the MAV or apply the CBF condition to a set of sample points on the collision envelope of the MAV. The latter could result in less conservative behaviour but incurs additional computation costs.

V. Evaluation

In order to evaluate our method, we performed several experiments both in simulation and on a real MAV. The MAV is teleoperated by a human operator. The teleoperation inputs consist of position and yaw references that drive the MAV from one position to another in the environment. We apply the proposed CBF safety filter to the raw position inputs provided by the teleoperator and generate certified position references that ensure *(i)* the MAV does not collide with any obstacles in the environment and *(ii)* the MAV does not enter areas that are not yet mapped. The yaw reference provides an additional degree of freedom for mapping the unknown environment but does not directly affect the safety of the MAV system. We therefore do not include the yaw reference in our safety filter design.

In both simulated and real-world experiments, OKVIS2 is configured to produce state estimates at 60 Hz, which are used by the position controller also operating at 60 Hz. The safety filter is implemented using qpOASES [28] and

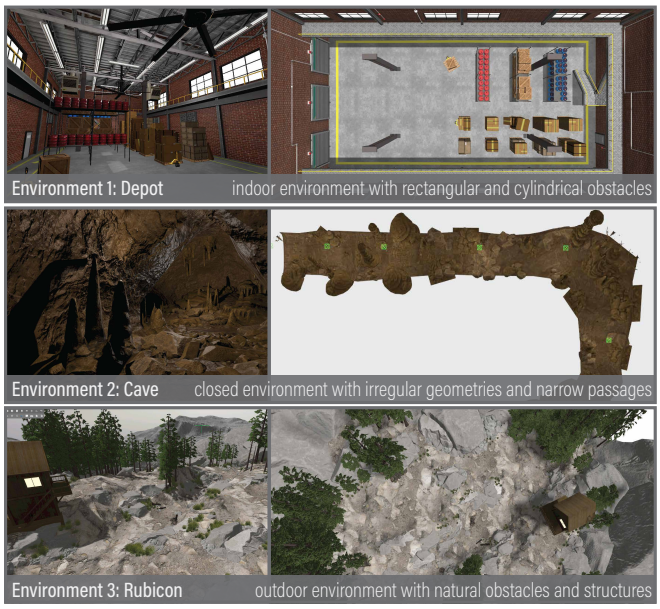

Fig. 4. Visualizations of the simulation environments used in our evaluation, each with different obstacle types and level of clutteredness [29].

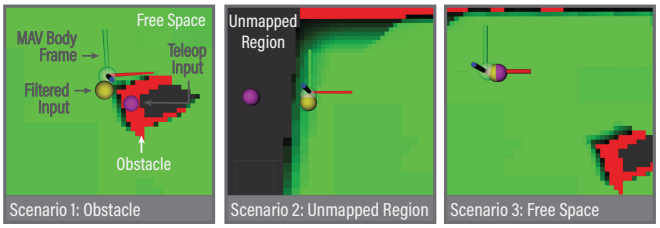

Fig. 5. Example cases from the evaluation in the Depot environment. In these plots, free space is indicated in green, obstacles are marked in red, and unknown area is shown in dark grey. The teleoperation input and the filtered input are indicated by purple and yellow spheres, respectively. The MAV position is drawn as a transparent sphere with its body frame attached. When the teleoperation input is unsafe—colliding with an obstacle [Left] or leading into the unmapped region [Middle], the safety filter finds the closest reference that does not violate safety constraints. In the interior of the safe region [Right], the filtered reference coincides with the teleoperation input.

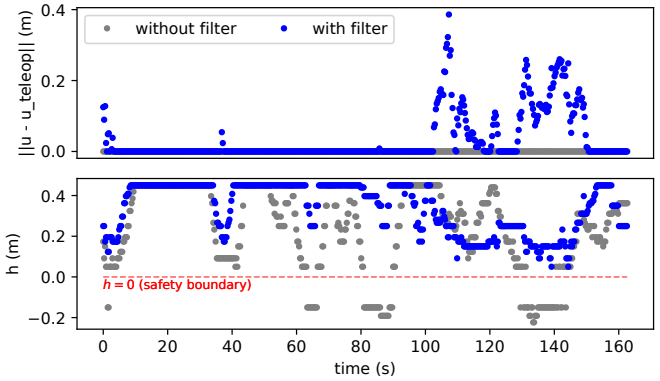

Fig. 6. Example trajectories illustrating the input correction applied by the safety filter and the corresponding CBF as the MAV is teleoperated in the Depot environment. The safety filter applies larger input corrections when the MAV is close to the safety boundary ($h = 0$) and applies approximately zero corrections when the MAV is in the free space (high h values). Without the safety filter, the teleoperated reference led to collisions at multiple time instances, while the filtered signals kept the MAV inside of the safe set.

is applied in real-time at 6 Hz. A map of the environment with a resolution of 0.05 m is updated using 640×480 depth images, and the truncation bound h_b for the TESDF is 0.5 m. In the CBF certification optimization problem, we use a slope of $p_1 = 0.45$ for the safe set to bound how fast the MAV is allowed to approach the safety boundary and a slope of $p_2 = 1 \times 10^{-3}$ for the unsafe region to define how quickly the MAV is required to converge to the safe region if it starts outside of the safe set. The noise parameter ε is set to 0.1 m in the simulation and 0.2 m in the experiment.

A. Simulation Results

We use the Gazebo [29] simulator with the PX4 autopilot [30] simulated in software. This results in a simulation with realistic MAV dynamics and sensors and a control interface that is exactly the same as the real-world MAV. The MAV model is based on the RMF-Owl [31], a $0.38 \times 0.38 \times 0.24$ m quadcopter, that is equipped with an Intel RealSense D455 RGB-D camera. We evaluate our approach in three different simulated environments with different obstacle types and levels of clutteredness (as depicted in Fig. 4).

Figure 5 illustrates the application of our proposed safety filter design. In the left and middle panels of the figure, we see that the raw references (depicted as magenta spheres) are directed toward either obstacles or unmapped regions. The filtered inputs (shown as yellow spheres) effectively constrain the robot (represented by the transparent sphere with the body frame attached) to remain within the safe region. Over the free space, as seen in the right panel, the safety filter does not modify the teleoperation input, and the MAV closely follows the raw teleoperation input. Fig. 6 shows a set of quantitative results corresponding to a test trial in the first simulation environment. When the MAV is close to the safety boundary ($h = 0$), the safety filter applies a larger correction to move the robot away from the boundary to prevent possible constraint violations and maintains approximately zero input correction over safe regions (with high h values). Over the course of this trajectory, the raw teleportation reference led to collisions at 83 discrete time instances, while the filtered reference led to zero constraint violation.

Table I presents a summary of our evaluation across three distinct simulation environments (Fig. 4), each comprised of three independent teleoperated trials. The trials corresponding to the teleoperated cases without the safety filter exhibited varying degrees of constraint violations, as indicated by the $h_{\min}$ values. Notably, The safety filter responded by applying larger input corrections, as quantified by $||\delta\bar{u}||$, in trials with more substantial constraint violations. Through the application of our safety filter, we have consistently achieved effective safety assurance for the teleoperated MAV system. The average adjustment in input introduced by the safety filter ranged from 0.03 m to 1.00 m. The $h_{\min}$ values spanned from -0.50 m to -0.15 m for teleoperated trials, while for trials with the safety filter, all $h_{\min}$ values are at least 0.05 m. This set of simulation evaluations demonstrates the efficacy

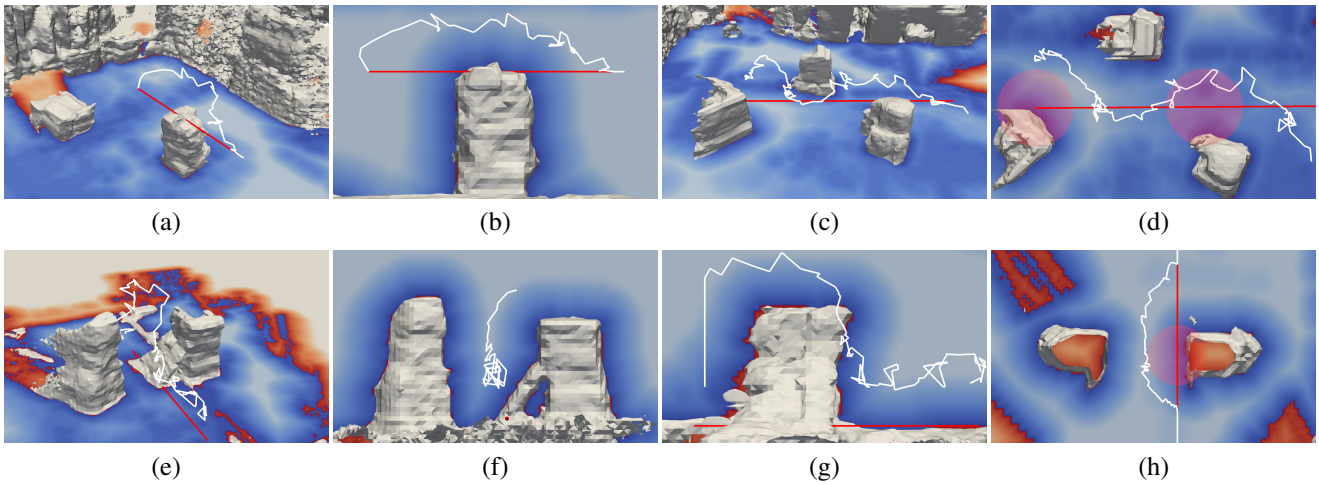

Fig. 7. Visualizations of the commanded (red) and filtered (white) reference position trajectories in real-world experiments. Positive (safe) TESDF values are shown in shades of blue, while negative (unsafe) values are in shades of red, as in Fig. 3 [Right]. Meshes extracted from the occupancy map are shown in gray, with some of the walls removed for clarity. Red circles with the MAV dimensions are shown to highlight some of the collisions that would have happened had the original reference been followed. (a), (b): Perspective and side view of an experiment where the commanded reference position passed through an obstacle. (c), (d): Perspective and top-down view of an experiment where the commanded reference position was too close to obstacles. (e), (f), (g): Perspective, front and side view of an experiment where the commanded reference position was too close to obstacles and the ground. (h): Top-down view of the experiment from Fig. 1 where the commanded reference position was too close to obstacles.

TABLE I

SUMMARY OF SIMULATION RESULTS

Env.	d	Teleop Input		Filtered Input		
		N_c	$h_{\min}$	N_c	$h_{\min}$	$\|\|\delta\bar{u}\|\|$
1	28	83	-0.22	0	0.05	0.03
	25	88	-0.49	0	0.05	0.23
	22	63	-0.50	0	0.05	1.00
2	10	43	-0.15	0	0.05	0.11
	13	34	-0.15	0	0.05	0.07
	24	11	-0.15	0	0.05	0.10
3	27	52	-0.15	0	0.05	0.04
	17	18	-0.15	0	0.09	0.04
	18	19	-0.19	0	0.05	0.04

Note: d is the distance traversed by the robot (in meters), N_c is the number of time steps with unsafe actions, $h_{\min}$ is the minimum value of CBF attained (in meters), and $||\delta\bar{u}||$ is the average input adjustment applied by the safety filter (in meters). Visualizations of the environments are shown in Fig. 4.

and robustness of our proposed perceptive safety filter for guaranteeing safe teleoperation in different environments.

B. Experimental Results

We performed experiments on a real MAV in order to showcase the applicability of our method in the real world. We use an MAV based on the Holybro S500 quadcopter frame equipped with an Intel RealSense D455 RGB-D camera and an NVIDIA Jetson Orin 16 GB computer, which measures $0.8 \times 0.8 \times 0.4$ m. All processing is performed on the MAV's onboard computer. It is worth noting that no motion capture system or prior knowledge of the environment is used for the experiments. A video of the experimental results can be found at http://tiny.cc/vi-slam-safe-filter.

To evaluate the versatility of our proposed safety filter approach, the MAV is teleoperated to navigate in four distinct environment configurations. Visualizations of the test environments, overlaid with segments of the teleoperated and filtered inputs, are presented in Fig. 7. Subfigures (a)-(b) illustrate the case where the safety filter adjusts the teleoperated reference (depicted in red) to enable the MAV to safely traverse over obstacles while maintaining adequate clearance (shown in white). In subfigures (c)-(d), the teleoperated reference is positioned in close proximity to obstacles, and the filtered reference guides the MAV through narrow passages without collisions. Subfigures (e)-(g) and (h) respectively correspond to the scenarios where the MAV encounters the ground plane and is required to maneuver under tight constraints. Notably, in subfigure (h), the safety filter enables the MAV to successfully pass underneath a bridge without any collisions (Fig. 1). These experimental results further validated the effectiveness of our safety filter in ensuring safe teleoperated MAV navigation across different scenarios and the applicability of our approach in a real-world setup where only onboard sensing and computation are accessible.

VI. CONCLUSION

In this work, we proposed a perceptive safety filter framework that seamlessly integrates CBF certification with VI-SLAM and dense 3D occupancy mapping for safe teleoperated navigation of MAVs. Our system updates the environment map and the CBF in real time, solely relying on the sensors and computation resources available to the MAV onboard. Through a series of simulations and experiments, we demonstrated that the safety filter can efficaciously detect and correct teleoperated reference inputs that would otherwise lead to safety constraint violations, highlighting its effectiveness in ensuring the safe operation of MAVs in unstructured and cluttered environments. As future work, we plan to further leverage the proposed framework for semantic exploration and evaluate the system in outdoor environments.

REFERENCES

[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3565–3572.

[2] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4666–4672.

[3] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[4] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1418–1425, 2018.

[5] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2019.

[6] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[8] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1689–1696.

[9] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[10] S. Leutenegger, "OKVIS2: Realtime scalable visual-inertial SLAM with loop closure," *arXiv preprint arXiv:2202.09199*, February 2022.

[11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011, pp. 127–136.

[12] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1241–1250, 2015.

[13] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. J. Kelly, and S. Leutenegger, "Efficient octree-based volumetric SLAM supporting signed-distance and occupancy mapping," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1144–1151, Apr. 2018.

[14] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean signed distance fields for on-board mav planning," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1366–1373.

[15] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[16] D. Duberg and P. Jensfelt, "UFOMap: An efficient probabilistic 3D mapping framework that embraces the unknown," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6411–6418, 2020.

[17] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental Euclidean distance fields for online motion planning of aerial robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4423–4430.

[18] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P. F. Alcantarilla, and S. Leutenegger, "Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3553–3560, April 2021.

[19] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[20] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking," *Journal of Field Robotics*, vol. 33, no. 1, pp. 133–152, 2016.

[21] F. Berkenkamp and A. P. Schoellig, "Safe and robust learning control with gaussian processes," in *Proc. of the European Control Conference (ECC)*, 2015, pp. 2496–2501.

[22] G. Joshi and G. Chowdhary, "Deep model reference adaptive control," in *Proc. of the IEEE Conference on Decision and Control (CDC)*, 2019, pp. 4601–4608.

[23] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Proc. of the Learning for Dynamics and Control Conference*, 2020, pp. 708–717.

[24] S. Dean, A. Taylor, R. Cosner, B. Recht, and A. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," in *Proc. of the Conference on Robot Learning*, 2021, pp. 654–670.

[25] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, "Barriernet: Differentiable control barrier functions for learning of safe robot control," *IEEE Transactions on Robotics*, 2023.

[26] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, 2023.

[27] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. of the American Control Conference (ACC)*, 2021, pp. 3882–3889.

[28] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.

[29] Open Robotics, "Gazebo Fortress," https://gazebosim.org.

[30] PX4 Autopilot, "PX4 Autopilot Software," https://px4.io.

[31] P. D. Petris, H. Nguyen, M. Dharmadhikari, M. Kulkarni, N. Khedekar, F. Mascarich, and K. Alexis, "RMF-Owl: A collision-tolerant flying robot for autonomous subterranean exploration," in *Proc. of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 536–543.