An Integrated Teleoperation Assistance System for Collision Avoidance of High-speed UAVs in Complex Environments

Conference Paper · June 2020

DOI: 10.1109/UR49135.2020.9144902

CITATION
1

READS
73

2 authors:

Min Wang
University of Luxembourg
5 PUBLICATIONS **25** CITATIONS

SEE PROFILE

Holger Voos
University of Luxembourg
240 PUBLICATIONS **1,569** CITATIONS

SEE PROFILE

An Integrated Teleoperation Assistance System for Collision Avoidance of High-speed UAVs in Complex Environments

Min Wang[1] Holger Voos[1]

***Abstract*— UAV teleoperation is a demanding task, especially for amateur operators who wish to successfully accomplish their mission without collision. In this work we present an integrated 2D LIDAR based Sense-and-Avoid system which actively assists unskilled human operator in obstacle avoidance, so that the operator can focus on higher-level decisions and global objectives in UAV applications such as search and rescue, farming etc. Specifically, with our perception-assistive vehicle control design, novel adaptive virtual cushion force field (AVCFF) based avoidance strategy, and integrated sensing solution, the proposed UAV teleoperation assistance system is capable of obstacle detection and tracking, as well as automatic avoidance in complex environment where both static and dynamic objects are present. The proposed system is constructed on the basis of Hector Quadrotor open source framework [1], and its effectiveness is demonstrated and validated on a realistic simulated UAV platform in Gazebo simulations where the UAV is operated at a high speed.**

I. Introduction

Recent years have seen rising interest in teleoperated Unmanned Aerial Vehicles (UAVs) from both research community and industry, as the maturity of fully autonomous UAVs is foreseen yet years to come. Teleoperated UAVs have been employed for many tasks, such as search and rescue, farming, entertainment and forestry etc. However, UAV teleoperation is a demanding task which requires skills and rich experience in order to ensure successful mission completion without collision. Nowadays highly-skilled UAV operators are usually sought after due to abundant UAV applications, which results in a shortage in supply. Additionally, it is usually expensive to hire those with high qualifications.

To address this pressing issue, we would like to develop a system which assists amateur operators to accomplish missions successfully in aforementioned tasks, where the UAV needs to safely maneuver in potentially unknown outdoor environment, due to operator's limited situation awareness. Therefore, in this paper we present a Sense-and-Avoid teleoperation assistance system that automatically detects and avoids collisions, so that the operator can focus on high-level decisions that serve his/her global objectives. During the course of avoidance, the proposed automatic assistance system also takes into account the intention of the operator, and would return to the intended control state after the avoidance maneuver.

*This work was supported by the "Fonds National de la Recherche" (FNR), Luxembourg, under the project C15/15/10484117 (BEST-RPAS).

[1]Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg {min.wang, holger.voos}@uni.lu

There have been only a few works that focus on UAV teleoperation assistance system in the literature. The majority focuses on autonomous UAV systems [2] [3] [4] [5]. Among the few existing relevant works, most of them deal with static environments, some even assume that the environment is fully known [6]. Rare works deal with complex unknown environments where there exist both static and dynamic obstacles. Furthermore, in their experiments and simulation UAVs are operated at very low speeds. For example, in [7], the UAV flies at the speed of $1.1m/s$, and in [8], it is at as low as $0.5m/s$. For a task which requires a rapid flight, such as rapid surveying of earthquake or flood area in search and rescue mission, our system provides a better alternative.

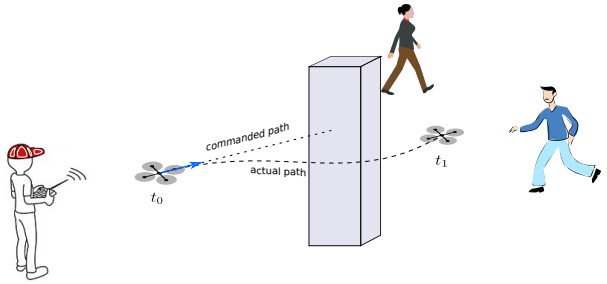

Fig. 1. Assisted UAV Teleoperation Scenario in Real World

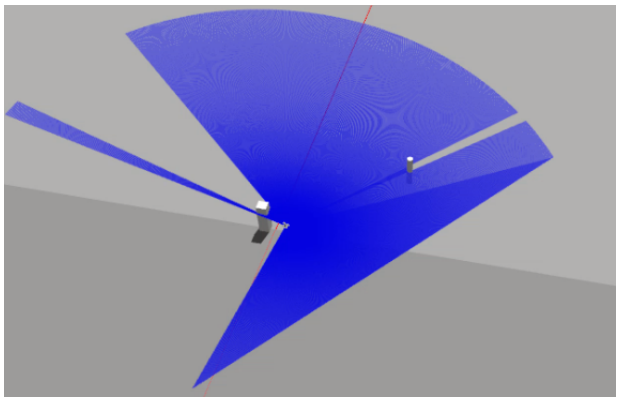

Fig. 2. Assisted UAV Teleoperation Scenario in Simulation

Our proposed system is constructed on the basis of Hector Quadrotor open source framework [1]. The advantage of this framework is that its simulated UAV has been validated against a real-world counterpart. In other words, after being validated on the simulated UAV, algorithms can be directly applied to a real-world UAV. The main contributions of this work are as follows: 1) We present in detail an integrated

2D-LIDAR-based Sense-and-Avoid framework for assisting UAV teleoperation; 2) We propose a PID-based forward-facing vehicle controller to assist perception; 3) We propose a novel adaptive virtual cushion force field (AVCFF) collision avoidance method to ensure fast and effective avoidance.

The rest of the paper is organized as follows: Related works are discussed in section II. Dynamics and vechivle control of the UAV platform employed in this work are introduced in section III, followed by a full elaboration on the proposed Sense-and-Avoid system in section IV. Simulation and results are discussed in section V. Finally, conclusions and future work are discussed in section VI.

II. Related Work

Different UAV teleoperation assistance systems for collision avoidance have been proposed in the literature with varying sensing modalities, environment representations and avoidance techniques.

[9] proposes a 2D LIDAR based optimization based Sense-and-Avoid solution which assists amateur human pilots in safe piloting in an environment where multiple moving objects exist. The proposed system, however, mainly targets at beginner-level operator who have near-zero UAV teleoperation skills or experience. Therefore, the operator's intention is not taken into account in this work. Additionally, in this work the sensing and avoidance subsystems are not yet integrated. Another 2D LIDAR based approach is introduced in [10], which proposes a model-based stochastic automatic collision avoidance algorithm for assisting UAV teleoperation. The proposed algorithm takes the pilot's input and exploits the robot's dynamics to predict the robot's trajectory for determining whether a collision will occur. It uses onboard sensor for obstacle detection, if a collision is imminent, the algorithm modifies the pilot's input to avoid the collision while attempting to maintain the pilot's intent. The environment is represented using line segments. Experiments are conducted in static indoor environment with walls.

An RGB-D camera based collision avoidance system is proposed in [8] to assist safe indoor navigation of teleoperated multirotor UAVs. The algorithm keeps track of detected obstacles in the local surroudings of the robot. The environment is represented with a set of cells in the near surrounding of the UAV. This representation is very similar to 3D occupancy grid. MPC is employed for generating avoidance control command, which modifies operator's input in order to avoid collision, while at the same time preserving the intent of the operator. Experiments are conducted in a static environment with a single object. [11] is an improvement of [8] with added functionalities.

Both [7] and [12] propose sonar-based teleoperation assistance systems for obstacle avoidance. [7] adopts MPC control technique to predict potential future collision, and uses SLAM and Octomap to construct a map of the environment. Simulation was conducted in a static environment with the UAV flies at a low speed. [12] uses FastSLAM to construct a rough occupancy grid map for the near sourroudings. The operator's input is overridden proportionally to an approximate TTC (time to collision), based on the vehicle's current velocity and distance to the obstacle. Depending on the level of threat, different response is given, such as no action, slow, stop, evasive maneuver. Simulation is carried out in a static indoor environment with walls.

[6] presents an automatic collision avoidance approach which assumes full environment awareness. The proposed method is essentially an MPC approach. The mechanisms of sensing, threat evaluation, and interaction between human operator and UAV are not discussed and assumed to be working. Experiment with real quadrotor is performed in a static lab environment, with the assumption of full environment awareness.

Among the few existing relevant works on UAV teleoperation assistance systems, model predictive control is found to be a popular avoidance technique. However, optimization-based algorithms are in general relatively slow in comparison to reactive avoidance approaches. Additionally, different perception sensor choices and world representations affect UAV's capability of perceiving and interpreting its sourrounding environment. In a high-speed flight, it is preferrable to choose a perception sensor with longer range of detection and better precision and fast update rate, thus 2D LIDAR is a reasonable option from this perspective.

III. Quadrotor Dynamics and Control

In this section, we briefly introduce the dynamics of the UAV platform we employ from Hector Quadrotor open source framework and its vehicle controller which we propose to facilitate forward-facing flight.

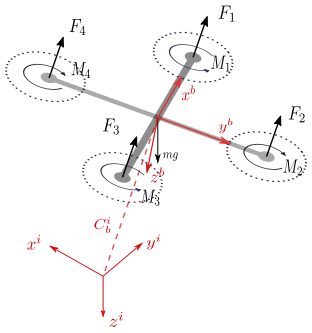

Fig. 3. Reference Frame Systems of Quadrotor

A. Dynamics of Quadrotor

The coordinate systems and frames of reference adopted are illustrated in Fig. 3, where inertial frames (upper index i) and body frames (uppper index b) are shown. The movement of the quadrotor can be described by the sum of all forces $\boldsymbol{F} = (F_1, F_2, F_3, F_4)^T$ and torques $\boldsymbol{M} = (M_1, M_2, M_3, M_4)^T$ acting on the vehicle in the following mathematical form:

$$\begin{aligned} \dot{\boldsymbol{p}}^i &= \boldsymbol{v}^i \\ \dot{\boldsymbol{v}}^i &= m^{-1}\boldsymbol{C}_b^i\boldsymbol{F} \\ \dot{\boldsymbol{\omega}}^b &= \boldsymbol{J}^{-1}\boldsymbol{M} \end{aligned} \tag{1}$$

where $\boldsymbol{p}^i$ and $\boldsymbol{v}^i$ are the position and velocity of the UAV's center of gravity in the inertial frame, $\boldsymbol{\omega}^b$ is its angular rate given in body frame, $\boldsymbol{C}_b^i$ is the rotation matrix that transforms coordinates from body frame to inertial frame, m represents mass of the UAV, and $\boldsymbol{J}$ is its inertia. For further details regarding dynamics of the UAV, please refer to [1] [13] [14].

B. Forward-facing Vehicle Control

Our previous work [9] introduces a novel nonlinear forward-facing flight controller which controls the UAV in such a way that the onboard perception sensor (in this case, 2D LIDAR) always perceives in the direction where the UAV flies, in order to ensure maximum situation awareness and thus safety. In this work, we adopt a similar concept, however, we create the forward-facing vehicle controller on the basis of the PID controller provided by the Hector Quadrotor framework.

In order to have a forward-facing flight controller in the existing framework, we modify the original vehicle controller (which we call lower-level controller), and cascade a higher level controller in front of the lower-level controller. Specifically, the higher-level controller takes in desired human operator velocity control commands $v^b_{x,hd}$ and $v^b_{y,hd}$, and generates control commands $v^b_{x,d}$ and ψ^b_d for the lower level controller, as shown in Fig. 4. The lower-level controller has been modified so that now it takes in yaw angle in body frame instead of that in inertial frame. The relationship between these two levels of controllers expressed in mathematical form is as follows:

$$v^b_{x,d} = f_1(v^b_{x,hd}, v^b_{y,hd}) = \sqrt{(v^b_{x,hd})^2 + (v^b_{y,hd})^2} \tag{2}$$

$$\psi^b_d = f_2(v^b_{x,hd}, v^b_{y,hd}) = \arctan \frac{v^b_{y,hd}}{v^b_{x,hd}} \tag{3}$$

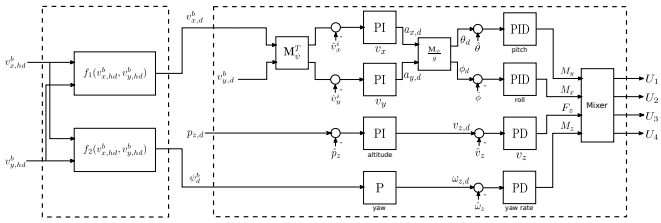

Fig. 4. Quadrotor Forward-facing Flight Controller Diagram

IV. Collision Avoidance System

Our collision avoidance system consists of three major functionalities, namely, environment perception, situation evaluation and collision avoidance. In the rest of this section, we will introduce them individually.

A. Environment Perception

We consider that environment can be modeled with cylindrical shapes of inifinite height, this assumption is valid in many aforementioned outdoor application scenarios. With this assumption, we could further simplify the model of the environment as a set of circles on the horizontal plane. The sensing functional module provides such an obstacle map (represented with circles) of the near sourroundings of the UAV utilizing an onboard 2D LIDAR. It consists of three major submodules, namely, scan preprocessing, shape generation and mapping and tracking, as shown in Fig. 5. In this paper we only outline the functionalities and techniques of each sensing submodule. For interested readers, please refer to [15] for more detailed explanation.

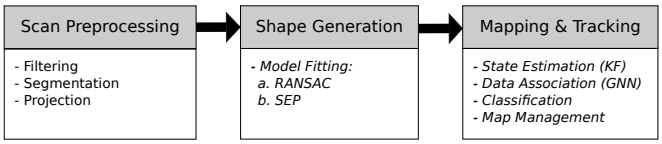

Fig. 5. Workflow of Sensing Functional Module

B. Situation Evaluation

The UAV is constantly evaluating the situation of its perceived surrounding to predict potential collisions. We incorporate two progressive situation evaluation policies to identify the level of collision threat posed by obstacles (or objects), namely, relative peripheral distance based evaluation and relative velocity based evaluation. The identified collision threat level, if any, will then be utilized later for adapting the avoidance strategy accordingly.

1) Evaluation based on relative distance: We adopt the safety zone set concept from [9], however, in this work the safety zone settings are modified to accomodate the challenges brought in by the high speed of UAV. The adapted safety zone set is illustrated in Fig. 6.

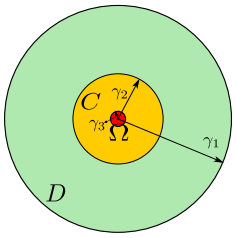

Fig. 6. Safety Zones for Collision Evaluation

For the purpose of illustration, some notations need to be defined. We define an integer set $\mathbb{Z} = \{1, 2, \ldots, n\}$ where n is the number of detected objects in the environment. $\boldsymbol{p}_u$ and $\boldsymbol{p}_{o_j}$ represent the x-y positions of the UAV and the jth object, i.e. $\boldsymbol{p}_u, \boldsymbol{p}_{o_j} \in \mathbb{R}^2$. We assume that the radius of detected object o_j is noted as λ_j. Additionally, d_{c_j} is defined as the distance between centers of the UAV and o_j, then its mathematical form is as follows:

$$d_{c_j} = \left\| \boldsymbol{p}_u - \boldsymbol{p}_{o_j} \right\| \tag{4}$$

The nearest area around the UAV is defined as Anti-object Zone $\boldsymbol{\Omega}$, indicated as red circular region in Fig. 6. This is the most safety-critical zone where no object should enter under any circumstance, because collision with the UAV would

take place once the zone is intruded. Anti-object Zone $\boldsymbol{\Omega}$ is defined as follows:

$$\boldsymbol{\Omega} = \{d_{c_j} \leq \gamma_3 + \lambda_j,\ j \in \mathbb{Z}\} \tag{5}$$

The next level of safety zone is defined as Conflict Zone, $\boldsymbol{C}$, as indicated in yellow in Fig. 6. Any object within this region of the UAV is considered as a potential collision threat and will trigger UAV's avoidance action.

$$\boldsymbol{C} = \{\gamma_3 + \lambda_j \leq d_{c_j} \leq \gamma_2 + \lambda_j,\ j \in \mathbb{Z}\} \tag{6}$$

Lastly, Detection Zone $\boldsymbol{D}$ as an exterior zone around the UAV is defined, as shown in green in Fig. 6. The UAV is merely observing the objects, and tries to establish tracking of objects and estimation of their velocities in this region for later use in case the objects come closer and violate another safety-level zone. Beyond zone $\boldsymbol{D}$, objects are considered non-existent to the UAV since they are beyond the detection range of the LIDAR sensor.

$$\boldsymbol{D} = \{\gamma_2 + \lambda_j \leq d_{c_j} \leq \gamma_1 + \lambda_j,\ j \in \mathbb{Z}\} \tag{7}$$

2) Evaluation based on relative velocity: If an object intrudes zone $\boldsymbol{C}$ of the UAV, then a further evaluation on the collision threat level is carried out, which is evaluating their relative velocity. Since we assume that the pose and velocity of the UAV are known, with the estimated velocity of the detected object from sensing module, we can calculate the estimated relative velocity between them. If their relative velocity is above a predefined velocity threshold V_{thres}, then we consider a higher level of threat is present, the collision avoidance functional module will generate a more aggressive avoidance action which ensures that the obstacle can be avoided in time.

C. Collision Avoidance

In this subsection, we present a novel adaptive virtual cushion force field (AVCFF) based control approach which generates avoidance maneuver commands to the UAV while preserving the operator's intent during the course of collision avoidance.

1) Adaptive Virtual Cushion Force Field: The proposed AVCFF method utilizes dynamic obstacle map generated from our sensing funcitonal module where each obstacle is represented as a cylinder (viewed as circle on x-y-plane), as well as evaluated threat level from the situation evaluatoin funcitonal module. The core idea of AVCFF method is that when the UAV encounters an obstacle intruding its conflict zone $\boldsymbol{C}$, as a collision avoidance strategy the UAV reacts as if it carries a circular safety cushion around it, that is, bouncing away from the obstacle due to the virtual repulsive force exerted on it, as illustrated in Fig. 7. More specifically, this virtual safety cushion is associated with an adaptive virtual force field function (which is detailed later in this section), which has zero influence on the UAV when the virtual cushion is not violated. However, when there are intrusions, it generates virtual forces to be exerted on the UAV in order to recover the virtual cushion from the deformation caused by

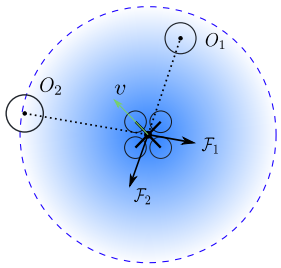

Fig. 7. Illustration of Adaptive Virtual Cushion Force Field (AVCFF) Collision Avoidance Method from Top-down View: o_1 and o_2 are two objects intruding the safety cushion of the UAV (i.e. zone $\boldsymbol{C}$), $\boldsymbol{\mathcal{F}}_1$ and $\boldsymbol{\mathcal{F}}_2$ are the resulting virtual forces due to the intrusion (sizes are not in proportion). The blue spreading shade provides an intuitive impression of the virtual force field function which determines the magnitude of virtual force field

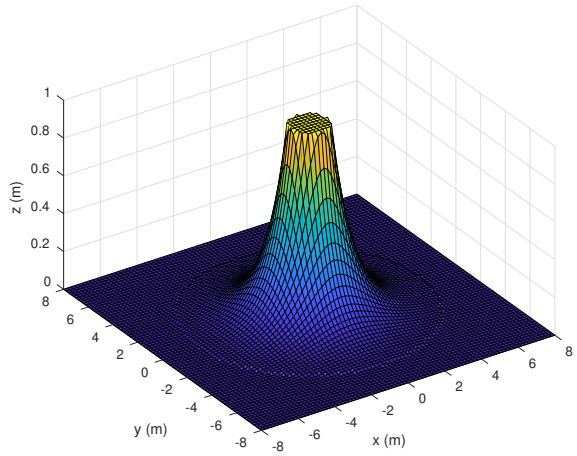

Fig. 8. Visualization of Virtual Force Field Function ($\mathcal{F}_{max} = 1$, $\mathcal{D}_{thres} = 6$, $d_{m_1} = \mathcal{D}_{min} = 1$)

obstacles. The UAV is then steered aways from the obstacles under the influence of the active virtual force field.

In traditional virtual force field method, such as potential field based collision avoidance method [16] [17] [18], where virtual force functions (i.e. potential functions) are constructed around the obstacles and goals, the robot is constantly under the influence of the combined virtual force field, and guided by it towards the goal.

The novelty of our method is that contrary to the traditional method, we construct the virtual force function around the robot (in our case, UAV) instead. Additionally, the virtual force generated by the virtual force function is adaptive based on the relative peripheral distance and relative velocity between the UAV and obstacle. The advantage of the proposed AVCFF collision avoidance method is that it is simpler in calculation and more time efficient as we do not need to calculate the gradient of potential functions to find the direction and magnitude of the virtual force. Additionally it is more suitable for real-world dynamic environment, as in a real-world scenario sensor noise would cause variation in the estimated cylindrical shapes of the obstacles, and would further results in constant changing of the potential functions around the obstacles, which is not desirable as it is time

consuming and not necessary.

The virtual force vector $\boldsymbol{\mathcal{F}}_j$ caused by object o_j is defined as

$$\boldsymbol{\mathcal{F}}_j = \mathcal{F}_j \cdot \boldsymbol{n}_j \tag{8}$$

where $\mathcal{F}_j$ is the magnitude of the virtual force, and $\boldsymbol{n}_j$ is its unit direction vector. The direction of the repulsive force $\boldsymbol{n}_j$ is defined as the direction pointing from the obstacle to the UAV, as indicated by the pointing direction of $\boldsymbol{\mathcal{F}}_1$ and $\boldsymbol{\mathcal{F}}_2$ in Fig. 7.

The adaptive virtual force function determines the magnitude of the virtual force, which is described in mathematical form as

$$\begin{aligned}\mathcal{F}_j &= f(d_{p_j}, d_{m_j}) \\ &= \begin{cases} \mathcal{F}_{max} \cdot e^{d_{m_j} - d_{p_j}} & \text{for } d_{p_j} \leq \mathcal{D}_{thres} \text{ and } \mathcal{F}_j \leq \mathcal{F}_{max} \\ \mathcal{F}_{max} & \text{for } d_{p_j} \leq \mathcal{D}_{thres} \text{ and } \mathcal{F}_j > \mathcal{F}_{max} \\ 0 & \text{for } d_{p_j} > \mathcal{D}_{thres} \end{cases}\end{aligned} \tag{9}$$

where $\mathcal{F}_{max}$ is a predefined maximum magnitude for any virtual repulsive force. $\mathcal{D}_{thres}$ is a constant representing the non-conflict peripheral distance threshold, and is defined by following equation:

$$\mathcal{D}_{thres} = \gamma_2 - \gamma_3 \tag{10}$$

d_{p_j} is the peripheral distance between the UAV and object o_j and is defined as

$$d_{p_j} = d_{c_j} - \lambda_j - \gamma_3 \tag{11}$$

Additionally, d_{m_j} defines the minimum peripheral distance at which the repulsive force reaches its maximum magnitude $\mathcal{F}_{max}$. The value of d_{m_j} is a function of the relative velocity between the UAV and the object o_j, as described below:

$$d_{m_j} = f(\hat{v}_{r_j}) = \begin{cases} \mathcal{D}_{thres} & \text{for } \hat{v}_{r_j} > \mathcal{D}_{thres} \\ \hat{v}_{r_j} & \text{for } V_{thres} \leq \hat{v}_{r_j} \leq \mathcal{D}_{thres} \\ \mathcal{D}_{min} & \text{for } others \end{cases} \tag{12}$$

where $\hat{v}_{r_j}$ represents the estimated relative velocity and V_{thres} is its predefined velocity threshold. $\mathcal{D}_{min}$ is a predefined constant, and is the absolute minimum peripheral distance that we would like to keep between the UAV and the object o_j during the entire course of avoiding this object, when the estimated relative velocity is below V_{thres}. In this case, the maximum repulsive force is reached when their peripheral distance is at $\mathcal{D}_{min}$. When the estimated relative velocity is bigger than V_{thres}, the function for generating the virtual force is adapted so as to drive the UAV to turn away faster from the obstacle. The value of d_{m_j} is reasonably capped by $\mathcal{D}_{thres}$.

Finally, the virtual forces caused by a total number of n objects are aggregated to form a total repulsive force:

$$\boldsymbol{\mathcal{F}}_{total} = \sum_{j=1}^{n} \boldsymbol{\mathcal{F}}_j \tag{13}$$

TABLE I

ALGORITHMIC PARAMETRIZATION FOR SIMULATED FLIGHT

Parameter	Value	Measurement Unit
γ_1	30	m
γ_2	6.3	m
γ_3	0.3	m
$\mathcal{D}_{thres}$	6	m
$\mathcal{F}_{max}$	5	N
$\mathcal{D}_{min}$	1	m
V_{thres}	1	m/s

2) Incorporation of Operator Intent: The final collision avoidance control command for the UAV, $\boldsymbol{v}^b_{avoid}$, takes into account both the generated repulsive force for collision avoidance and the pilot's operation intent.

The moment when the situation evaluation functional module identifies that potential collision threat exists and triggers generation of virtual force field, the most recently received control command from the human operator is memorized and converted into world frame, which is utilized during the course of avoidance as the intention velocity control command, noted as $\boldsymbol{v}^w_{intent}$.

The total repulsive force from Eq. (13) is translated into repulsive force velocity control command, $\boldsymbol{v}^w_{rpl}$. Their mathematical relationship is as follows:

$$\boldsymbol{v}^w_{rpl} = f(\boldsymbol{\mathcal{F}}_{total}) = \boldsymbol{\mathcal{F}}_{total} \tag{14}$$

With $\boldsymbol{v}^w_{intent}$ and $\boldsymbol{v}^w_{rpl}$, we can obtain $\boldsymbol{v}^b_{avoid}$ with following equations:

$$\boldsymbol{v}^b_{avoid} = \boldsymbol{R}^b_w(\boldsymbol{v}^w_{intent} + \boldsymbol{v}^w_{rpl}) \tag{15}$$

Finally, the direction of $\boldsymbol{v}^b_{avoid}$ will be used to guide the UAV's velocity, while the magnitude of the velocity remains constant. This combination is then used as control input for the controller displayed in Fig. 4, completely overriding the pilot's new control input during the course of collision avoidance. The course of collision avoidance is considered complete when the UAV detects no objects in its zone $\boldsymbol{D}$ and flies at the velocity of the operator's intent, $\boldsymbol{v}^w_{intent}$. We assume that at this point in time the UAV would send a signal to the remote control to trigger it to slightly vibrate/or make a sound or light so as to warn the human operator that the control over the UAV will be returned.

V. SIMULATION AND RESULTS

We conducted simulation experiments using ROS [19] and Gazebo [20] simulator. Since the simulated UAV model provided by Hector Quadrotor framework has been validated against its real-world counterpart, thus we believe that the results obtained from our proposed algorithms in these simulation experiments are highly comparable to real-world results.

A. Simulation Scenario and Setup

The simulation scenario consists of one static cubical object o_1 with size $1 \times 1 \times 3m$, one moving cylindrical object o_2 with size $0.3 \times 3m$, and one quadrotor UAV

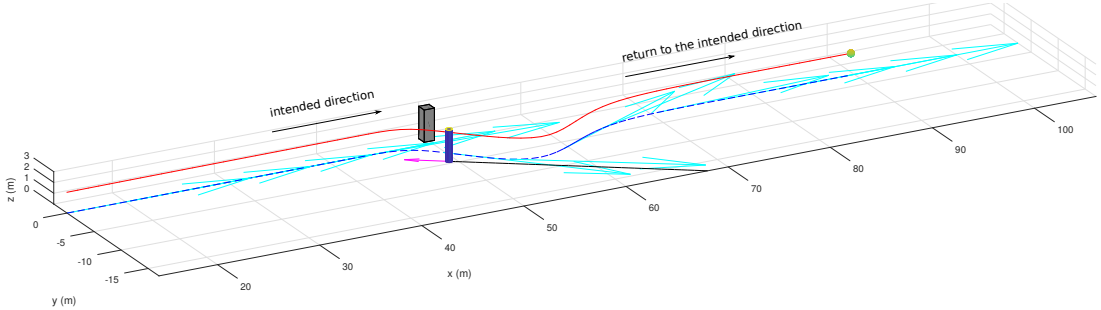

Fig. 9. 3D View of Simulation Scenario: The red solid line represents traveled 3D trajectory of the UAV, the blue dashed line is its projection onto the ground, the cyan arrows attached to the blue dashed line represent projected velocity directions at different locations and time instances. The greenish sphere attached to one end of the red line represents the UAV. The black line is the projected trajectory of the moving cylindrical obstacle o_2, the magent arrow attached the black line stands for the velocity direction of o_2. The cubical shape is the static obstacle o_1

which is deemed as a sphere with radius γ_3. From top-down view, object o_1 stands at x-y-location $(50, 1)m$, the moving object o_2 moves in a direction towards o_1 and the UAV with constant velocity $(-1.2, 0.8)m/s$, and the UAV is constantly commanded to move forward with a high speed $5m/s$ towards both objects, for example, as shown in Fig. 9. The moving object is designed in such a way that it mimics a person in walking speed. During the whole flight, the UAV is maintained at a certian height (in this case $2m$) by an altitude controller.

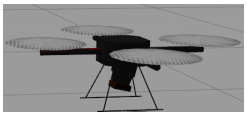

Fig. 10. Simulated UAV with Adapted LIDAR

Additionally, our proposed collision avoidance system has been parametrized as in Table I. Furthermore, we have pitched the 2D LIDAR onboard the simulated UAV by $0.2rad$ so that the LIDAR faces slightly upwards (the short red line on one arm of the UAV represents the front), as shown in Fig. 10. Since during a high-speed flight, a realistic UAV would constantly have a pitching angle in order to maintain the speed, which undesiredly results in that the onboard parallelly-placed LIDAR constantly facing the ground other than the front. Therefore, it is necessary to make this adjustment to ensure effective perception.

B. Results Discussion

With the aforementioned simulation scenario, we run the simulation experiments 20 times, however, the UAV is commanded to take off at different timing so as to vary the timing and locations in which the UAV encounters the obstacles. The experiments results are shown in Table II. Among the 20 simulated flights, only one flight failed which is caused by slow response of the UAV during dramatic turning. This issue can be potentially solved by tuning the parameters of the avoidance algorithm, which we will investigate further in future research. Among the successful 19 flights, the shortest peripheral distance between the UAV and obstacle is $0.64m$, which is sufficient distance to avoid obstacles.

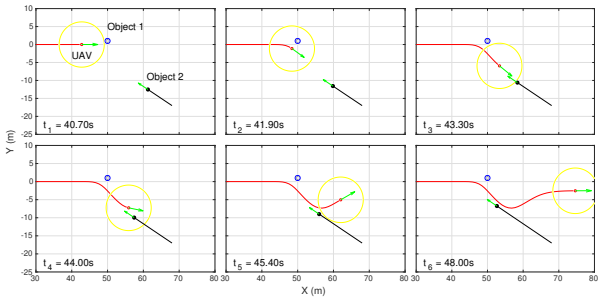

Fig. 11. Simulated Flight from Top-down View: solid lines represent traveled trajectories (red for UAV, black for moving object o_2). The red circle visualizes the exterier border of minimum safety space for the UAV, so is blue circle for static object o_1 and black circle for object o_2. The green arrows represent the velocity directions of respective subjects at the locations and time instances

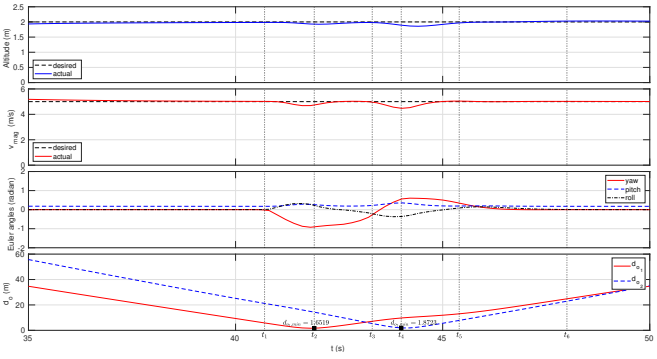

Fig. 12. Numerical Analysis of Simulated Flight

In the following we detail one of the successful experiments, whose 3D view is as depicted in Fig. 9. Several salient moments of the flight are captured, as shown in Fig. 11. Illustration of the meaning of different colors and shapes used can be found in the caption of Fig. 11, while the evolution of the Sense-and-Avoid process presented in the subfigures is detailed as follows:

Subfigure at time $t_1 = 40.70s$ displays the moment when the UAV starts to detect the presence of static object o_1 in its conflict zone $\boldsymbol{C}$, and decides that avoidance action is necessary. It memorizes the most recently received control

TABLE II
RESULTS OF MULTIPLE FLIGHTS

No. of Flights	Success Rate	Shortest Peripheral Distance
20	95%	$0.644m$

command from the human operator as intent velocity command $\boldsymbol{v}^b_{intent}$. Additionally, the intrusion of o_1 starts to cause generation of a virtual force field which subsequently acts on the UAV to deviate it from the originally intended path if the UAV purely follows the intention velocity command $\boldsymbol{v}^b_{intent}$, so as to avoid collision. Subfigure at time $t_2 = 41.90s$ depicts the moment when the UAV is at the closest location to o_1 during the course of avoidance. From Fig. 12, we can see that their peripheral distance at this point is $d_{o_1,min} = 1.6519m$. Subfigure at time $t_3 = 43.30s$ shows the moment when the UAV starts to detect the presence of the approaching object o_2 in zone $\boldsymbol{C}$ shortly after o_1 is out of it. Therefore, o_2 triggers again the generation of virtual force field which redirects the UAV to avoid collision with o_2. Subfigure at time $t_4 = 44.00s$ depicts the moment when the UAV is at the closest location to o_2 during the course of avoidance. From Fig. 12, we can see that their peripheral distance at this point is $d_{o_2,min} = 1.8723m$. Subfigure at time $t_5 = 45.40s$ describes the moment when o_2 is just out of zone $\boldsymbol{C}$ of the UAV. From now on the UAV has no obstacles in sight, i.e. in both zone $\boldsymbol{D}$ and $\boldsymbol{C}$, it starts to restore its flight direction to the state before the avoidance course takes place. Subfigure at time $t_6 = 48.00s$ depicts the moment when the UAV has fully returned back to the intended velocity $\boldsymbol{v}^b_{intent}$.

We assume that after successfully returning back to the last commanded velocity (intent velocity), the UAV would send a signal to RC to trigger it to slightly vibrate or make a sound so that the human operator is notified that the control of the UAV has been returned.

Fig. 12 provides us with a numerical anaysis of the simulated flight. From the first subfigure, we can see that the altitude controller is functioning well in maintaining the desired altitude. The control command from the operator has consistantly been $5m/s$ forward in the body frame. Therefore at time t_1, the Sense-and-Avoid system of the UAV takes this command as its intended velocity. During the whole course of collision avoidance, the UAV tries to maintain its velocity magnitude v_{mag} constant at $5m/s$, though from the second subfigure in Fig. 12 we see that between t_1 and t_5 there are small deviations from the desired value, which is caused by the rapid avoidance manoeuvre, which we could infer from the third and fourth subfigures. The UAV is, however, capable of rapidly restoring the desired velocity by adjusting the pitch angle accordingly, as shown in the Euler angle subfigure. Additionally, as the UAV steers its moving direction by yawing, for a realistic UAV it inevitably causes small roll movements, also as shown in the Euler angle subfigure.

In our submitted supplementary video, we present the simulated flight in Gazebo and its corresponding visualization in rviz. In rviz, the created map consists of detected obstacles (circles) is visualized, as well as the generated virtual force caused by intrusion of the obstacles. The first scenario in the video is the corresponding simulation for the results discussed in this section. The second scenario in the video presents a more challenging situation.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a 2D LIDAR based UAV teleoperation assistance system to automate the collision avoidance task for the operator so that he/she can consentrate on higher-level decisions and global objectives during teleoperation. Our proposed assistance system is constructed on the basis of the well known Hector Quadrotor open source framework, and incorporates a perception-assistive forward-facing vehicle controller, a 2D LIDAR obstacle detection and tracking sensing solution, a progressive situation evaluation policy, and a novel AVCFF-based reactive avoidance method. With all of these components working in harmony, the effectiveness of the proposed assistive automatic collision avoidance system is demonstrated and validated on an realistic simulated UAV platform in several Gazebo simulations, where UAV is operated in high speed in complex envirionment with static and moving objects.

As future work, we plan to further validate proposed algorithm on a real-world UAV in different and even more challenging scenarios. Additionally, we will compare proposed collision avoidance methods against other reactive avoidance approaches quantitively, in order to gain a more comprehensive view of the proposed solution.

REFERENCES

[1] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. Von Stryk, "Comprehensive simulation of quadrotor uavs using ros and gazebo," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2012, pp. 400–411.

[2] T. Marinho, M. Amrouche, V. Cichella, D. Stipanović, and N. Hovakimyan, "Guaranteed collision avoidance based on line-of-sight angle and time-to-collision," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4305–4310.

[3] O. Cetin and G. Yilmaz, "Real-time autonomous uav formation flight with collision and obstacle avoidance in unknown environment," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 415–433, 2016.

[4] R. Sharma, J. B. Saunders, and R. W. Beard, "Reactive path planning for micro air vehicles using bearing-only measurements," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 409–416, 2012.

[5] S. Hrabar, "Reactive obstacle avoidance for rotorcraft uavs," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4967–4974.

[6] J. Israelsen, M. Beall, D. Bareiss, D. Stuart, E. Keeney, and J. van den Berg, "Automatic collision avoidance for manually tele-operated unmanned aerial vehicles," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6638–6643.

[7] B. Giovanini, H. A. Oliveira, and P. F. Rosa, "An automatic collision avoidance approach to assist remotely operated quadrotors," in *International Conference on Intelligent Autonomous Systems*. Springer, 2016, pp. 213–225.

[8] M. Odelga, P. Stegagno, and H. H. Blthoff, "Obstacle detection, tracking and avoidance for a teleoperated uav," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2984–2990.

[9] M. Wang and H. Voos, "Safer uav piloting: A robust sense-and-avoid solution for remotely piloted quadrotor uavs in complex environments," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019.

[10] D. Bareiss, J. R. Bourne, and K. K. Leang, "On-board model-based automatic collision avoidance: application in remotely-piloted unmanned aerial vehicles," *Autonomous Robots*, vol. 41, no. 7, pp. 1539–1554, 2017.

[11] M. Odelga, P. Stegagno, N. Kochanek, and H. H. Bülthoff, "A self-contained teleoperated quadrotor: On-board state-estimation and indoor obstacle avoidance," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7840–7847.

[12] J. Mendes and R. Ventura, "Assisted teleoperation of quadcopters using obstacle avoidance," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 7, no. 1, pp. 54–58, 2013.

[13] S. A. Raza and W. Gueaieb, "Intelligent flight control of an autonomous quadrotor," in *Motion Control*. IntechOpen, 2010.

[14] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*. Springer, 2017, vol. 118.

[15] M. Wang, H. Voos, and D. Su, "Robust online obstacle detection and tracking for collision-free navigation of multirotor uavs in complex environments," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 1228–1234.

[16] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, Sep. 1989.

[17] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, March 1985, pp. 500–505.

[18] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE, 1991, pp. 1398–1404.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[20] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.