A General Framework for Human-Drone Interaction under Limited On-board Sensing

S. Nayhouse, S. Chadha, P. Hourican, C. Moore, N. Bezzo, *Member, IEEE*

***Abstract*— Recent advancements in unmanned aerial vehicles (UAVs), has allowed their deployment for numerous applications like aerial photography, infrastructure inspection, search and rescue, and surveillance. Despite the potential for full autonomy, many applications still necessitate human operators for navigating complex environments and decision-making. Existing solutions often employ high-precision and simple sensors like 2-D or 3-D LiDAR, which may provide more data than necessary and contribute to increased system complexity and cost. To address these challenges and bridge the gap between full autonomy and human-controlled UAVs, this work develops a shared-autonomy framework for UAVs, leveraging lightweight, low-cost 1-D LiDAR sensors combined with mobility behaviors to obtain performance comparable to more advanced 2-D/3-D LiDAR sensors while minimizing energy, computation overhead, and weight. Our framework includes a novel state machine method that exploits the UAV mobility to compensate for the limitations of 1-D LiDAR sensors, ensuring safety and obstacle avoidance through a physics-based algorithm that transitions between teleoperation and autonomous mode as needed based on environmental conditions and safety-critical issues. Experimental validations on real UAVs demonstrates the effectiveness of this shared autonomy scheme in complex environments, and the system is further generalized to larger UAVs and prototyped with a custom sensor configuration and onboard obstacle avoidance.**

I. Introduction

Flying is the dream of many but only a few can truly experience it freely. What was once seen as science fiction, however, is now becoming a reality as new advances in computation and sensing lead to the development of advanced robotic systems. Autonomous mobile robots are entering society and finding many applications like aerial photography, infrastructure inspection, surveillance, hobby application, and even search and rescue operations. Unmanned aerial vehicles (UAV), a "class of aircraft that can fly without the onboard presence of pilots" [1], have sparked debates among experts about the advantages and disadvantages of fully autonomous systems compared to human-controlled aerial vehicles.

Although fully autonomous UAVs hold great promise, drone crashes, safety challenges, security issues, and privacy concerns have led many to question whether the risks and costs associated with these systems outweigh their benefits. In response to these concerns, this research project aims to address the limitations of fully autonomous UAVs by developing a shared autonomy (or semi-autonomous) system that combines the benefits of human-controlled input and onboard autonomy, while minimizing sensing overhead using simpler sensors and leveraging mobility to gather more data.

The primary aim of this project is to develop a safer and more efficient drone that reduces crash likelihood due to operator error, crucial for advancing autonomous UAV technology. Combining human control and onboard autonomy, we balance operator inputs with safety and obstacle avoidance. In this work we deal with a specific type of UAV, the quadrotor, composed of four rotors on a rigid body [2], offering maneuverability, stability, and a convenient development platform. Our shared autonomy approach ensures safe teleoperation in cluttered environments, even with unseen obstacles, using lightweight 1-D sensors for safe navigation without requiring complex 3D sensors. With two interchangeable techniques—reactive navigation without yaw and yaw-based mapping—we avoid collisions and maintain safe distances, adapting to real-time UAV dynamics.

This project addresses challenges associated with fully autonomous UAVs while retaining their benefits by seamlessly integrating human input, onboard autonomy, and strategic use of simpler sensors and UAV mobility. The shared autonomy system provides a practical middle ground between fully autonomous and human-controlled UAVs, enhancing safety, efficiency, and performance, and paving the way for broader adoption of this transformative technology.

This paper is organized as follows: in Section II, we provide an overview of related work, discussing the existing literature and techniques in the field of UAV autonomy and obstacle avoidance; in Section III, we formally define the problem and describe the challenges and goals of our research; in Section IV, we present our approach, detailing the methods and algorithms used to achieve shared autonomy and safe navigation; finally, in Section V, we discuss the conclusions drawn from our work and explore possible avenues for future research and improvements.

*Research supported by CoStar and the Mead Endowment program.

S. Nayhouse is with the departments of Systems and Information Engineering and Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: san6yhk@virginia.edu).

S. Chadha is with the department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: sc3xry@virginia.edu).

P. Hourican is with the department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: pjh4as@virginia.edu).

C. Moore is with the department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA (email: cm5jx@virginia.edu)

N. Bezzo is with the departments of Systems and Information Engineering and Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA (email: nb6be@virginia.edu)

II. RELATED WORK

The concept of shared autonomy has been extensively studied in the context of UAVs and other robotic systems. Our novel approach simultaneously considers human inputs and onboard sensors/autonomy for obstacle avoidance and task completion, unlike previous projects that used manual or autonomous modes separately.

Early attempts at shared autonomy in UAVs, such as the system proposed by [3], allowed switching between teleoperation and autonomous modes but lacked concurrent processing of human inputs and onboard autonomy for obstacle avoidance. Our method addresses this limitation by incorporating both inputs simultaneously.

[4] developed a shared autonomy framework for quadrotor UAVs using onboard vision sensors for obstacle detection and avoidance. However, their approach relied on switching between teleoperation and autonomous modes, leading to suboptimal performance in complex environments. Our approach integrates human input with onboard autonomy for more efficient and safer operation.

In assistive robotics, [5] investigated shared autonomy for robotic wheelchairs, considering user input and onboard sensing data for navigation. Our project focuses on UAVs, addressing unique challenges such as obstacle avoidance and dynamic flight environments.

Limited sensing navigation has been explored in various studies, emphasizing simpler sensors like 1-D LiDAR to reduce complexity and computational demands. [6] demonstrated the feasibility of obstacle avoidance and navigation using 1-D LiDAR sensors and a reactive control algorithm. Our method similarly minimizes sensing overhead by leveraging simpler sensors and exploiting the UAV's mobility to gather more data.

To summarize, existing shared autonomy systems for UAVs often require the operator to switch between manual and autonomous modes, leading to potential inefficiencies and safety concerns. Our proposed method advances the field by enabling the concurrent use of human inputs and onboard autonomy for obstacle avoidance and task completion, while also employing limited sensing navigation.

III. PROBLEM STATEMENT

In this project, we aim to develop a shared autonomy system for UAVs that seamlessly integrates human inputs with onboard autonomy, allowing the drone to be teleoperated by a human while safely navigating cluttered environments and maintaining a certain distance from obstacles, even when the human operator cannot see or does not notice them. Formally:

A. Problem – UAV Shared Autonomy in Cluttered Environments

Consider a quadrotor UAV tasked to be teleoperated by a human while avoiding N_o obstacles positioned at $o_i = [x_{o,i}, y_{o,i}, z_{o,i}]'$, with $i = 1, ..., N_o$, along its followed trajectory τ in a cluttered environment. The quadrotor operation is considered safe and successful if $\|x(t) - o_i\| > \varepsilon$, $\forall_t > 0$ and $\forall_i = 1,...,N_o$, where ε is a minimum distance to maintain that considers the dimension of the quadrotor and $\|\cdot\|$ is the Euclidean distance norm.

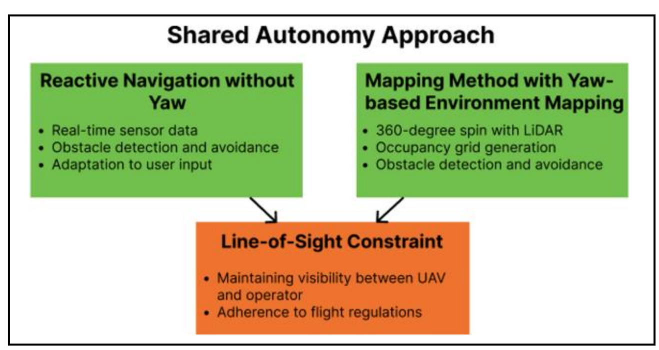
The shared autonomous system is designed to simultaneously process manual inputs from a human operator, $u_h(t)$, and onboard autonomy data, $u_a(t)$. The objective is to find a control policy $\pi(x(t), u_h(t), u_a(t))$ that guides the UAV along the trajectory τ while maintaining a safe distance from obstacles during teleoperation, even if the human operator cannot see or does not notice the obstacles. The system should satisfy the following conditions:

- Safety: Ensure UAV maintains safe distance from obstacles ($\|x(t) - o_i\| > \varepsilon$, $\forall_t > 0$ and $\forall_i = 1,...,N_o$).
- Teleoperation: Allow human operators to control the UAV while the control policy $\pi(x(t), u_h(t), u_a(t))$ ensures safety and obstacle avoidance.
- Shared autonomy: The control policy $\pi(x(t), u_h(t), u_a(t))$ integrates human inputs and onboard autonomy for safe teleoperation in cluttered environments.
- Adaptive obstacle avoidance: The policy should maintain a minimum distance ε from moving obstacles by adjusting the UAV's trajectory.
- Limited sensing capabilities: Despite constraints like sparse sensors, system must ensure safe and efficient operation by leveraging available sensor data for obstacle detection, avoidance, and navigation.

IV. APPROACH

Fig. 1 presents our framework for the shared autonomy approach to enable UAVs to navigate cluttered environments while being teleoperated by a human operator. The approach consists of a state machine with three components: two interchangeable obstacle avoidance techniques (reactive navigation without yaw and mapping method with yaw-based environment mapping) and a line-of-sight constraint that can be included independently of the other two components. These techniques focus on avoiding collisions and maintaining a safe distance from obstacles, even if the human operator cannot see or doesn't notice the obstacles.

Figure 1. State Machine Framework of Shared Autonomy Approach



A. N 1-D LiDAR Sensors

The first component of our shared autonomy scheme involves the use of N 1-D LiDAR sensors mounted on the UAV in various configurations, such as a single sensor, four sensors pointing in the cardinal directions, or other configurations. These sensors gather data during the UAV's flight and map the surrounding environment using a point cloud representation. Let S_i represent the sensor data from the i-th LiDAR sensor, where $i \in \{1, 2, ..., N\}$. The data are processed to generate a point cloud representation P, with each point $p_j \in P$ defined as $p_j = (x_j, y_j, z_j)$, representing the coordinates of an obstacle in the environment.

1) Reactive Navigation Method

The reactive navigation method is a comprehensive approach designed to ensure safe and efficient UAV navigation in cluttered environments by combining obstacle detection and avoidance with user-guided navigation. This method addresses the limitations of obstacle detection and avoidance by incorporating the user's input for a more controlled and intuitive navigation experience while maintaining safe distances from obstacles and ensuring collision-free operation.

In this obstacle detection and avoidance technique, the quadrotor uses available sensors to detect obstacles and plan in obstacle-free directions. The collision avoidance algorithm employs a reactive navigation approach, incorporating the drone's dynamics and constraints, such as maximum velocity and acceleration. This method calculates the repulsive forces F_{rep} from detected obstacles in the point cloud P. Given the current position $x(t)$ and velocity $v(t)$ of the UAV, the repulsive forces are calculated as:

$$F_{rep}(p_j) = k\left(\frac{1}{\left|\left|x(t) - p_j\right|\right|} - \frac{1}{\varepsilon}\right)\left(\frac{1}{\left|\left|x(t) - p_j\right|\right|^2}\right)(x(t) - p_j)$$

where k is a positive constant, ε is the minimum safe distance, and $\|x(t) - p_j\|$ is the distance between the UAV and the obstacle point p_j with $j \in \{1, 2, ..., N_j\}$. The total repulsive force F_{total} is the sum of the repulsive forces from all detected obstacles:

$$F_{total} = \sum_{j=1}^{N_j} F_{rep}(p_j)$$

The control input u_h is adjusted to avoid obstacles by adding the total repulsive force:

$$u_{adj} = u_h + F_{total}$$

Although the reactive navigation method is adaptable to dynamic and unpredictable situations, its efficiency may be limited, as it continuously adjusts the UAV's trajectory in response to obstacles. Furthermore, the system may not detect all obstacles due to limited sensors, leading to potential blind spots or suboptimal paths. To address these challenges, user-guided navigation is integrated into the reactive navigation strategy. This approach builds upon the obstacle detection and avoidance by tailoring the navigation strategy based on the user's input. This allows for more controlled navigation that adapts to the user's intended direction while still maintaining safe distances from obstacles.

For example, if the UAV is close to the left wall but the user is moving forward, the system can accept that command until the vehicle gets too close to the wall ahead. At this point, the system triggers a fixed command in the direction toward the left or right, depending on which one has the larger distance, and maintains that direction until an opening is found or the UAV gets too close to an obstacle.

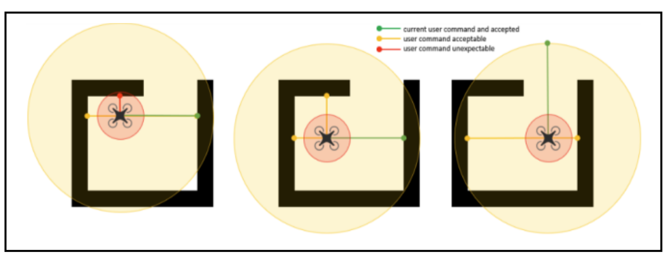
In this updated strategy, the control input u_h is adjusted based on the user's command, denoted as u_{comm}, and the minimum safe distance to the wall ahead, represented by ε_{wall}. The system then calculates the adjusted control input u_{adj}, considering the distance to obstacles in the user command direction and triggering a fixed command to move in the direction with the most significant clearance when the UAV gets too close to an obstacle:

$$u_{adj} = \begin{cases} u_{comm} & if\ \|x(t) - p_{wall}\| \geq \varepsilon_{wall} \\ argmax(\Delta d_{left}, \Delta d_{right})u_{max} & if\ \|x(t) - p_{wall}\| < \varepsilon_{wall} \end{cases}$$

where $x(t)$ is the current position of the UAV, p_{wall} is the position of the wall ahead, Δd_{left} and Δd_{right} are the distances to the left and right walls, respectively, and u_{max} is the maximum allowable control input. When the condition is met, the system triggers a fixed command u_{fixed} in the direction toward the left or right depending on which one has the larger distance, then calculates the adjusted control input u_{adj}. While the example above is considering a vehicle commanded to move forward, it can be generalized to other commanded directions in a similar fashion. The method considers obstacle distance in the command direction and, if too close, issues a fixed command to move toward the largest clearance.

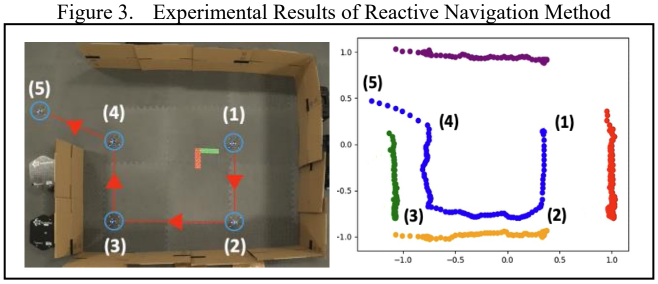
By developing a reactive navigation method that seamlessly integrates obstacle detection and avoidance with user-guided navigation, the system effectively addresses the challenges associated with fully autonomous UAVs while maintaining their potential benefits. This combination of techniques results in a more robust and versatile navigation system that ensures safe, efficient, and intuitive UAV operation in complex environments. Fig. 2 shows how our reactive navigation policy adjusts its command based on user-commanded control input as well as obstacle location information sensed by the 1-D LiDAR sensors.

Figure 2. Simulated Drone Movement in Reactive Navigation Method



To validate the effectiveness of this strategy, we conducted a series of experiments in real-life scenarios with UAV flights. The quadrotor used in these experiments is a Crazyflie mini-UAV with on-board 4 1D range sensors facing forward, backward, left, and right. The following figures visually represent the performance of our reactive navigation strategy.

Figure 3. Experimental Results of Reactive Navigation Method



The subfigure on the right in Fig. 3 shows the point cloud data for the four sensors on the quadrotor (forward=green, backwards=red, left=yellow, and magenta=right) while the blue dotted line represents the path followed by the quadrotor and measured by a high precision motion capture system. The experimental results, as depicted in the figures, demonstrate the effectiveness of the proposed strategy in real-life scenarios with data collected from UAV flights. The figures show the UAV successfully navigating through various environments while employing the updated control input adjustment strategy. These illustrations indicate that the UAV maintains a safe distance from obstacles and adapts its trajectory based on the user's input. Furthermore, the results reveal that the drone can effectively avoid collisions by switching to a fixed command when approaching walls or obstacles, moving in the direction with the most significant clearance. In summary, the figures provide strong evidence of the practicality and robustness of our proposed strategy, showcasing its ability to ensure the safety and effectiveness of UAV operations in complex and cluttered environments.

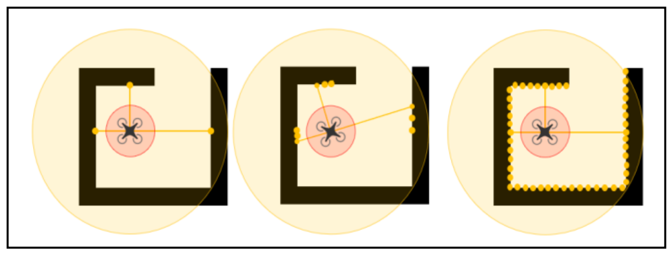
2) *Mapping Method (Yaw-based Environment Mapping)*

In this technique, the drone performs a 360-degree spin using the 1-D LiDAR sensors to create a probabilistic occupancy grid map of the environment. The LiDAR sensor data are used to update the occupancy grid *G*, representing the environment with cells *G(x, y, z)*. Instead of marking cells as occupied or free, the grid cells store log-odds values, updating it based on the point cloud *P* received from sensors.

We use a traditional *log-odds_update* function which computes the change in log-odds for a given cell based on new sensor data in *P* at every iteration. As more data are collected, the log-odds values in the grid cells become more accurate, reflecting a higher certainty about the presence or absence of obstacles in the environment. This allows the human teleoperator to fly the drone freely while the drone autonomously avoids collisions and maintains a safe distance from obstacles in the mapped environment. The control input u_h is adjusted accordingly based on the occupancy grid and obstacle locations to ensure obstacle avoidance. When the drone leaves the mapped area, it must perform a subsequent 360-degree spin using the LiDAR sensors to update the map of the environment before the human can pilot it again 4to guarantee safety. The algorithm for the yaw-based mapping is visually showed in Fig. 4 and detailed below:

1. Initialize occupancy grid map *G* and log-odds values for the specific range sensors installed on the UAV.
2. Perform a 360-degree spin of the drone to collect LiDAR sensor data and generate the point cloud *P*.
3. For each point *(x, y, z)* in the point cloud *P*:
    1. Update the occupancy grid *G* using the *log-odds_update* function.
4. Adjust the control input u_h based on the occupancy grid *G* and obstacle locations to ensure obstacle avoidance.
5. If the drone leaves the mapped area, perform steps 2-4 to update the map of the environment.

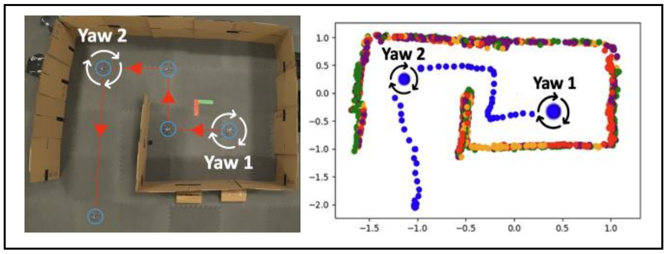
Figure 4. Simulated Drone Movement in Yaw-Based Mapping Method



This technique creates a detailed map for a comprehensive understanding of the environment, eliminating blind spots due to limited sensor configurations. The extensive mapping allows the human operator to navigate the drone safely and freely, while facilitating optimal path planning for obstacle avoidance using the complete map. After building the map, the drone uses its sensors to continuously verify data consistency with the map, akin to a particle filter. Inconsistencies prompt a remapping process to update environmental knowledge. However, there are drawbacks. This technique requires an initial and intermittent 360-degree yaw to generate and update the map, which can be time-consuming, especially in time-critical situations. Periodic remapping might be necessary when the drone moves to a new area outside the initial map's field of view, leading to further delays. While efficient in known environments, this method may be less adaptable to dynamic or rapidly changing situations due to its reliance on the initial map's accuracy.

To validate the effectiveness of the mapping method, we have conducted a series of experiments in real-life scenarios with UAV flights. The following figures visually represent the performance of our yaw-based environment mapping strategy. By examining these figures, we aim to provide compelling evidence of our strategy's practicality and robustness, highlighting its potential to enhance the safety of UAV operations in complex environments.

Figure 5. Experimental Results of Yaw-Based Mapping Method



The experimental results, as depicted in the figures, demonstrate the effectiveness of the proposed mapping method in real-life scenarios with data collected from UAV flights. As for the previous case, the subfigure on the right shows the point cloud data collected by the four sensors on the quadrotor. Since the quadrotor is performing a yaw motion to map the environment, the point cloud map has mixed colors since as the same obstacles are captured at least 4 times during a 360 revolution. The figures show the UAV successfully navigating through various environments while employing the yaw-based environment mapping strategy. The results reveal that the drone can effectively avoid collisions and plan more optimal paths to escape obstacles by utilizing the comprehensive environmental understanding provided by the mapping method.

B. Line of Sight Constraint

The third component of our approach, independent of the other two, incorporates the location of the human pilot, the obstacle, and the drone to determine the line of sight. This method adheres to the rules of flight that require a line of sight between the operator and the UAV, as mandated by the Federal Aviation Administration (FAA) to ensure safe operation and prevent potential collisions [7]. We assume that the user has some localization shared with the UAV. Maintaining line of sight not only complies with FAA regulations but also enables the human operator to provide more accurate and context-aware guidance to the UAV, enhancing the overall effectiveness of the shared autonomy system. We calculate the line of sight (LOS) constraint:

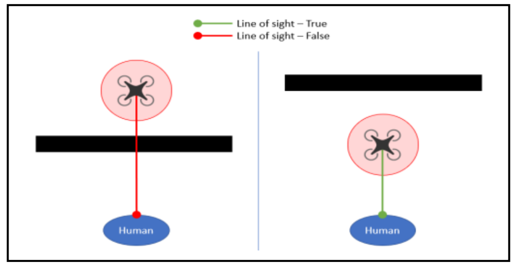
$$LOS = True, if\ \forall o_i \in O, \frac{||(x(t) - x_h) \times (x_h - o_i)||}{||x(t) - x_h||} > \varepsilon\ ;$$

$$LOS = False, otherwise$$

where x_h is the human operator's position, $x(t)$ is the UAV's position at time t, $l(t)$ is the line connecting x_h and $x(t)$, O is the set of all objects in the environment, o_i is an obstacle where $o_i \in O$, ε is the minimum safe distance from an obstacle, and the Line of Sight (LOS) status is either True or False. If False, the drone is not in the direct line of sight of the human (i.e., line between the drone and the human intersects an obstacle), and the system restricts the pilot's control until the line of sight is reestablished. The algorithm for the line of sight (LOS) constraint is detailed and visualized in Fig. 6.

1. Obtain position of human operator (x_h) and UAV ($x(t)$) and calculate the line connecting the human operator and the UAV ($l(t)$).
2. For each obstacle o_i in the environment, calculate the distance between the line $l(t)$ and the obstacle.
3. Check if the calculated distance is greater than the minimum safe distance (ε) for all obstacles.
4. If the calculated distance is greater than ε for all obstacles, set LOS to True; else, set LOS to False.
5. If LOS is False, restrict the human operator's control until LOS is reestablished.

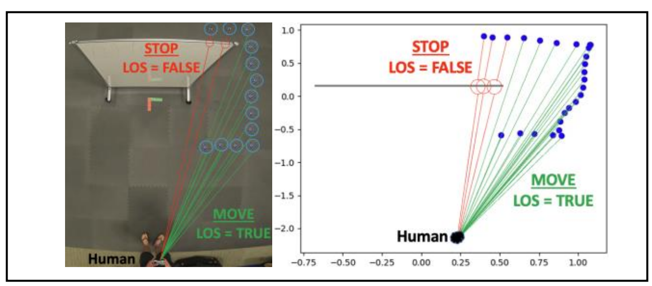
Figure 6. Simulated Drone with Line of Sight Constraint



This method enhances safety by ensuring that the human operator always has a clear view of the drone's position and surroundings. For simplicity, we make the UAV stop and hover in place when the line of sight is lost. However, a more robust approach could involve moving the drone toward a line-of-sight state (or towards the last previous position where the line of sight was established).

To validate the effectiveness of the line-of-sight constraint method, we have conducted a series of experiments in real-life scenarios with UAV flights. The following figures visually represent the performance of our line-of-sight constraint strategy. By examining these figures, we aim to provide compelling evidence of our strategy's practicality and robustness, highlighting its potential to enhance the safety and effectiveness of UAV operations in complex and cluttered environments.

Figure 7. Experimental Results of Line of Sight Constraint



The experimental results in the figures show the proposed line of sight constraint method's effectiveness using real-life UAV flight data. They depict the UAV maintaining line of sight with the operator while navigating different environments. These illustrations indicate that the UAV adheres to the line of sight rules and restricts the pilot's control when the line of sight is obstructed by obstacles. Furthermore, the results reveal that the drone can effectively reestablish line of sight by stopping and hovering in place, and the line of sight can only be reestablished if the human operator moves to a new position with a clear view of the drone. In summary, the figures demonstrate the practicality and robustness of the proposed method, ensuring safe and effective UAV operations in complex settings.

C. *Extension to Other UAV Systems*

To evaluate our reactive navigation method, we developed a prototype semi-autonomous drone for autonomous obstacle avoidance using live LIDAR sensor data. Starting with a non-autonomous drone controlled by a joystick, we incorporated our design for a semi-autonomous version. In this configuration, a human tele-operator steers the drone using a joystick, while four LIDAR sensors continually monitor the surroundings. If a nearby obstacle is detected, the drone autonomously avoids it using reactive navigation and resumes its course once the obstacle is clear of specified boundaries.

To detect objects, we utilized four blind-spot LIDAR sensors mounted on a 3D tower affixed to the drone and connected to a USB hub, which conveys the data to our microcontroller. The software subsequently analyzes the data and performs the necessary obstacle detection and avoidance. Our software methodology consists of Python scripts managing reactive navigation using Terabee EvoRanger sensors and ROS topics. A ROSVars class maintains sensor readings and updates joystick inputs. The main loop initializes nodes for sensor and joystick readings, publishes drone velocity, and sets a 1000 loop rate for optimal performance. If the joystick button isn't pressed (as a fail-safe), the program processes sensor readings, adjusting the drone's direction and outputting messages reflecting current operation.

To confirm the effectiveness of our software in processing sensor readings for realistic aerial vehicle scenarios, we utilized the Gazebo simulator and real-world experiments. The simulation allowed the drone to take off, hover at a predetermined altitude, maintain its x and y positions, and safely descend upon program termination. After successful simulation, a physical experiment was conducted in an actual environment to further validate the drone's performance. The drone effectively replicated its simulated actions, showcasing autonomous navigation capabilities by avoiding obstacles and responding to joystick inputs, demonstrating the reliability and effectiveness of our software methodology in real-life applications. Please see [8] for more information.

Figure 8. Holybro Drone with Applied Shared Autonomy System

V. CONCLUSION AND FUTURE WORK

This research introduces a shared-autonomy framework for UAVs, bridging the gap between fully autonomous and human-controlled systems. Using lightweight 1-D LiDAR sensors and mobility behaviors, the framework achieves comparable performance to advanced 2-D/3-D LiDAR sensors with reduced energy consumption, computation, and weight. The shared autonomy approach ensures safety and obstacle avoidance by transitioning between teleoperation and autonomous mode. Experimental validation in real-world environments demonstrates the scheme's effectiveness, enhancing UAV safety, efficiency, and performance.

This research on shared autonomy for UAVs is crucial for various stakeholders, including manufacturers, developers, researchers, and industries that depend on UAVs. Future research could explore combining reactive navigation without yaw and yaw-based mapping, potentially creating a more robust and versatile system. A continuous yawing strategy while the UAV moves could enable real-time, high-resolution mapping, improving obstacle anticipation and avoidance.

Advancements in UAV technology and shared autonomy systems can significantly enhance safety, efficiency, and overall performance, paving the way for broader adoption of this transformative technology. Continued progress in shared autonomy will empower individuals and industries to unlock the full potential of UAV technology, ultimately benefitting the world.

ACKNOWLEDGMENT

The authors thank Pravardhan Nagireddy, Jacob Higgins, and Lauren Bramblett for their technical assistance and guidance throughout this project.

REFERENCES

[1] Federal Aviation Administration (FAA), "Subtitle B—Unmanned Aircraft Systems," FAA, 2022. www.faa.gov/uas/resources/policy_library/media/Sec_336_UAS.pdf

[2] A. Yel and N. Bezzo, "Assured Runtime Monitoring and Planning: Toward Verification of Neural Networks for Safe Autonomous Operations," IEEE Robotics & Automation Magazine, 27(2), 102-116

[3] M. A. Goodrich, J. W. Crandall, and E. R. Andersen, "Teleoperation and Beyond for Personal Aerial Vehicles," in Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction, ser. HRI '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 67–74.

[4] L. Chen, K. M. Wurm, S. H. Tomic, J. Hertzberg, and T. Zhang, "LiDAR-IMU Time Delay Calibration in Robot Localization and Mapping," 2016 IROS, Daejeon, South Korea, 2016, pp. 4792-4798.

[5] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared Autonomy via Hindsight Optimization for Teleoperation and Teaming," The International Journal of Robotics Research, vol. 37, no. 7, pp. 717-742, Jun. 2018.

[6] J. Pérez, V. Milanes, and E. Onieva, "Cascade architecture for lateral control in autonomous vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 2040-2050, Oct. 2014.

[7] Federal Aviation Administration (FAA), "Unmanned Aircraft Systems (UAS) Regulations & Policies," https://www.faa.gov/uas/regulations_policies/

[8] S. Nayhouse, "Autonomous Obstacle Avoidance for UAVs," YouTube, https://youtu.be/wol1z0ap23c