2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall) | 979-8-3315-1778-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-Fall63153.2024.10757896

Advancing Autonomous UAVs: Safe Navigation and Object Avoidance in Dynamic Airspace

Gavin S. Halford, Logan P. McCorkendale, Zachary J. McCorkendale, Eliana E. Jaques,
Kamesh Namuduri, and Colleen P. Bailey
Department of Electrical Engineering, University of North Texas, Denton, TX, USA

***Abstract*—Autonomous Unmanned Aerial Vehicles (UAV) hold the potential to revolutionize logistics and transportation. To become truly viable, this technology must prove its capability to operate safely across a wide range of environments and conditions. Factors like wind, rain, hail, birds, and the presence of other drones in the airspace must all be considered in the decision-making process. While traditional control systems struggle with the complexity of this problem, machine learning has shown promise in tackling these challenges efficiently and effectively. This work proposes to advance independent drone operation through object avoidance, data collection, and smart navigation.**

I. Introduction

In the rapidly evolving landscape of autonomous technology, UAVs are leading a significant shift in logistics and transportation. As fully autonomous operation becomes increasingly feasible, the primary concern remains their ability to navigate and operate safely within complex and dynamic environments. This paper aims to develop an automated system for UAV operation that prioritizes operational capability, decision-making, and adaptability to unforeseen challenges.

Real-time object detection and avoidance are crucial components of any autonomous vehicle navigation system. Operating within designated air corridors, drones must swiftly react to unexpected obstacles such as birds, other drones, and airborne debris. Effective response demands that UAVs not only detect and track these potential hazards but also adeptly maneuver around them. Traditional approaches often rely on LIDAR and the Vector Field Histogram (VFH) model for obstacle detection and avoidance [1]. In contrast, this paper introduces a energy-efficient alternative that utilizes millimeter-wave radar for enhanced performance in UAV applications. This approach provides a more efficient and lightweight solution, significantly improving obstacle detection and avoidance capabilities in dynamic environments.

UAVs are vulnerable to adverse weather, such as wind and rain, which can lead to equipment loss and ground safety risks. Weather conditions can change rapidly and vary along air corridors, making monitoring at fixed spots insufficient. UAVs should function as both navigators and data collectors, gathering weather data for real-time analysis. This paper proposes an on-drone system that collects weather and location data, transmitting it to a ground station where real-time constraint boundaries are set for conditions beyond operational limits. This system improves navigational accuracy and helps manage risks in autonomous aerial traffic management.

Long-range planning and navigation based on in-flight data analysis enable an autonomous UAV to anticipate and adapt to dynamic environmental conditions effectively. This proactive capability allows the UAV to reroute in response to emerging challenges such as impending storms or to avoid congested areas, enhancing both operational efficiency and overall safety.

Such adaptability is pivotal for the reliability of autonomous UAV logistics. Previous studies have demonstrated the effectiveness of reinforced learning models for navigation and traffic control, as seen in the work of Hwang et al. [2], which explored adaptive control strategies in dynamic environments, and Kodama et al. [3], which focused on congestion management through intelligent routing algorithms. Building on these foundations, this paper proposes integrating a Deep Q-Network with real-time data collection to develop a more sophisticated navigation and air corridor traffic management system. This approach aims to improve decision-making processes under varying aerial traffic conditions, thereby advancing the state-of-the-art in autonomous UAV operations. Building on these foundations, this paper proposes integrating a Deep Q-Network with real-time data collection to develop a more sophisticated navigation and air corridor traffic management system. This approach aims to improve decision-making processes under varying aerial traffic conditions, providing a state-of-the-art solution for autonomous UAV operations.

A. Main Contributions

1) Introduction of a millimeter-wave radar-based object detection and avoidance system, enhancing UAV navigation efficiency and safety.
2) Development of an on-drone weather and location data collection system that establishes real-time constraint boundaries, improving risk management in dynamic weather conditions.
3) Integration of a Deep Q-Network with real-time data for proactive navigation and traffic management, advancing the decision-making capabilities of autonomous UAVs.

II. Methodology

A. Object Detection and Tracking

The object detection system is built around a Texas Instruments 1843 millimeter wave radar module, equipped with three transmit and four receive antennas. Although LIDAR has traditionally been favored for UAV detection and ranging due to its high accuracy and resolution, it often under-performs in inclement weather and is usually bulky and power-hungry. This project prioritized energy efficiency and adverse weather performance, making millimeter wave radar the preferred depth sensor. Configured via the Texas Instruments MM-Wave Visualizer tool, the module offers a 50-meter maximum range, with adjustable settings for range, resolution, and velocity [4]. Its built-in CFAR algorithm filters out transient and reflective signals to enhance accuracy.

Paired with a USB camera using the YOLOv4 [5] object detection algorithm trained on the Microsoft COCO dataset [6], bounding boxes around detected objects are mapped to the vehicle plane and fused with RADAR data using camera intrinsics calculated with MATLAB's Camera Calibration Toolbox.

The camera and RADAR are aligned in an over/under setup, with the camera above the RADAR antennas. Static detections are filtered out, and the data is merged using a DBSCAN clustering algorithm. A Joint Probabilistic Data Association (JPDA) algorithm assigns detections to tracks using their probability density functions, while movements are predicted with a constant velocity extended Kalman filter. Finally, a Global Nearest Neighbors (GNN) tracker and Kalman filter was applied to the YOLO bounding boxes prior to visualization to ensure a more robust visual tracking system.

B. Object Avoidance

A key feature for RADAR sensors is that they have the ability to use the Doppler effect to return velocity vector information from detected objects. This project prioritized avoidance methods that could harness this RADAR characteristic. The object avoidance algorithm combines Velocity Obstacle (VO) [7] and Artificial Potential Fields (APF) [8] approaches. The VO algorithm identifies potential collisions by comparing the UAV's velocity vector to those of detected objects. If on a collision course, a collision cone is calculated, and the algorithm finds avoidance vectors outside this cone, selecting the one that minimally deviates from the current course. This minimizes significant course corrections, but the VO algorithm has limitations when objects enter the field too rapidly or shallowly, or when complex vectors can't be solved.

To address these issues, the APF algorithm provides a backup. It assigns attractive and repulsive weights to maintain safe distances between detected objects and the UAV, while guiding the UAV towards a target direction. Although less efficient than VO, APF effectively maintains separation. Coding and simulation were done in MATLAB.

C. Data Collection

The Calypso Instruments Ultra-Low-Power Ultrasonic Wind Meter (ULP STD) was selected as the wind speed and velocity vector sensor. It uses four ultrasonic transducers for data collection and supports various communication protocols like MODBUS, RS485, UART, I2C, and analog 4-20mA [9]. The UART protocol was chosen for this design due to low power consumption (0.15mA at 38,400 baud, 0.45mA at 115,200 baud) at 5V. The anemometer was configured using the manufacturer's software to transmit data at 115,200 baud and 1Hz, aligning with the telemetry protocol. Data is collected via the COM port and integrated with telemetry for Ground Control Station (GCS) transmission.

D. Smart Navigation

Smart navigation uses a Deep Q-Network (DQN) reinforcement learning model. The DQN processes WGS-84 coordinate waypoints and designated weather avoidance areas from a UAV flight plan. It translates these points into a pixel grid maze that guides the UAV (agent) from the starting to the final waypoint while avoiding obstacles.

Built with Python and TensorFlow, the DQN consists of six descending dense layers, ending with a layer of size four representing possible agent movements: left, right, up, and down. A 2D grid serves as input, where 0 marks empty squares, +1 is the agent, -1 is the goal, and -2 are obstacles. During training, a Python script randomizes the agent, obstacle, and goal positions to enhance dynamic input adaptability.

Rewards are given based on movement quality: +0.1 for moves closer to the goal and -0.1 for each step to minimize unnecessary steps. An obstacle collision earns -1, and moving outside the grid is -0.5. Reaching the goal rewards 100 points, ending the process. Adjustments in reward values ensure optimal outcomes. The DQN ultimately outputs a set of movements translated back to WGS-84 coordinates to update the UAV's course.

III. Results

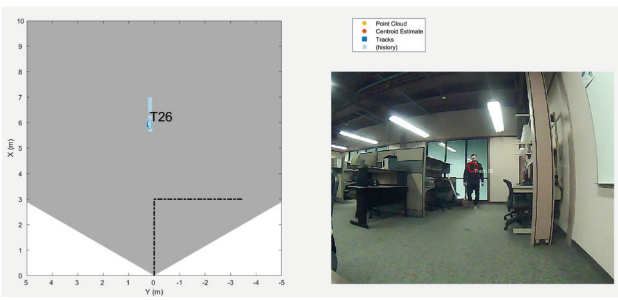
A. Object Detection and Tracking

Sensor testing involved visualizing tracks and measuring distances. A theater plot using MATLAB's RADAR Toolbox provided a bird's-eye view of tracks and detections on the right, while camera detections were shown as bounding boxes. Tracks were projected back into the camera plane as red circles, each with a track identifier to verify sensor alignment. Due to the dynamic nature and many adjustable parameters of this fused sensor system, design and testing were closely linked.

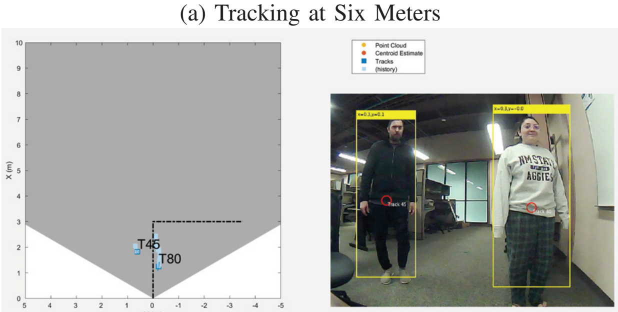
After initial tuning of individual sensors, track-to-track, direct detection, and clustered detection fusion methods were explored. Track-to-track fusion was dismissed because translating successful camera tracks into the vehicle plane and then combining them with RADAR tracks induced excessive lag in the tracker. Thus, detections were combined earlier through concatenation and DBSCAN clustering. GNN and JPDA multi-object trackers were then tested and adjusted for each sensor setup.

The RADAR paired best with JPDA, while the camera performed best with GNN. However, when camera detections were translated to the XY plane, distance accuracy declined with changes in bounding box size. This was mitigated by assigning higher noise values to camera distance data, reducing their impact on object tracking. The final fused tracker used RADAR and XY-translated camera detections combined with DBSCAN clustering. A JPDA tracker with an extended Kalman filter constant velocity model turned these clustered detections into tracks.

For visualization, a GNN tracker and constant velocity Kalman filter were applied to YOLO camera detections, projecting them back as bounding boxes. The final sensor model visualization confirmed distance accuracy (shown in Fig. 4b), and demonstrated successful multi-object tracking capabilities (shown in Fig. 4c). While prototype testing was performed using people as the tracked object, the sensor will be used for anticipated in air objects (birds, drones, etc.).



(a) Tracking at Six Meters



(b) Two Object Tracking

Fig. 1: Object Detection and Tracking

B. *Obstacle Avoidance*

The obstacle avoidance algorithm, designed in MATLAB and tested in a simulation environment, utilized a representative detection and tracking system. A UAV followed an elliptical flight path at an average speed of 1.2 m/s (2.25 MPH). Multiple groups of avoidance objects moved along randomized paths within a set range, with individual velocities ranging from 0.83 to 4.45 m/s (1.86 to 9.95 MPH). The simulation's maximum avoidance velocity was 5.65 m/s (12.2 MPH). Randomized obstacle trajectories intersected the UAV's path at varying angles, speeds, and group formations, allowing for diverse testing scenarios.

Design and testing were blended for algorithm tuning. In Fig. 2, the simulation UAV (green) picks up an object with collision potential that is approaching quickly at a perpendicular angle (Fig. 4a). The VO algorithm starts making small smooth direction changes. An imaginary solution is calculated and discarded (Fig. 4b). Finally, the object hits the safety radius and the APF algorithm finishes the avoidance maneuver (Fig. 4c) and successfully evades the object (Fig. 4d).

C. *Data Collection*

The anemometer was calibrated using the approved manufacturer software and communication protocols were tested. Data from the wind sensor was successfully transmitted to the GCS in real time. A CesiumJS web application was developed to visualize the wind constraints based on the transmitted data. The wind data was combined with location data, and as wind is simulated from the sensor, constraints are visualized in the Cesium environment based on preset polygon size values.

D. *Deep Q-Network*

Training the Deep Q-Network (DQN) was an incremental process, with navigational complexity increasing as the model improved. Initially, training began on an empty grid with only the agent and the goal to fine-tune the reward system. The initial reward system gave positive points for progress toward the goal and reaching it, while a -1 penalty was assigned for moving away. This setup worked for simple navigation but hindered exploration as the environment became more complex.

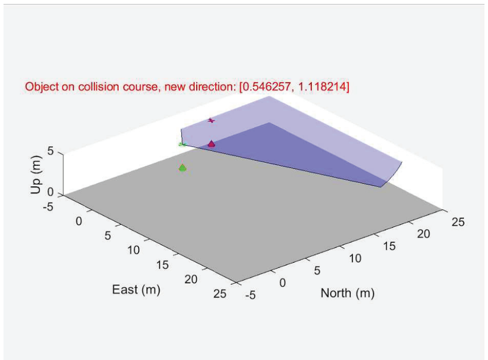
After the DQN consistently reached the goal, an obstacle was added for complexity. However, the DQN exploited a loophole where moving onto an obstacle only incurred a standard penalty and then vacated the spot, allowing it to ignore obstacles later. This flaw was fixed by reinstating the obstacle in its original position. The reward system was also adjusted, reducing rewards for movement and imposing stricter penalties for collisions.

The model then learned to handle multiple obstacles but achieved lower scores because it didn't prioritize minimizing steps. A negative reward for each step, balanced with positive feedback for effective moves, helped the DQN improve navigation through more complex paths. The final score earned for each episode evaluates the performance of the model, where lower scores are representative of an efficient route to the endpoint. In Fig. 3a the model performance for the initial rewards structure is consistently below zero and has an erratic training graph. After tuning, model performance is much more efficient and consistent as shown in Fig. 3b. Due to the random generation of obstacles, instances of the agent becoming trapped within the grid occasionally occurred and are characterized by the sharp drops seen in Fig. 3b.

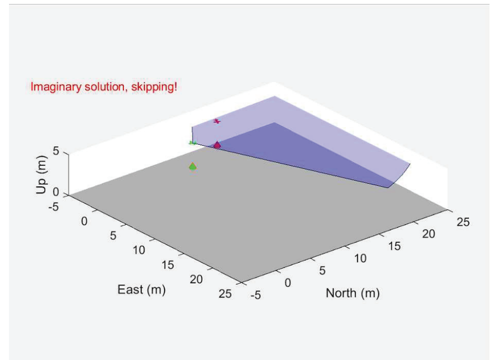
E. *Drone Software Integration and Visualization*

Due to safety concerns, this project's initial scope was confined to simulation. A Software-In-The-Loop (SITL) drone was created using PX4 software within the Gazebo simulator to test the interaction between the avoidance algorithm and UAV flight control. The drone's flight plan was visualized using Q-Ground Control (QGC).

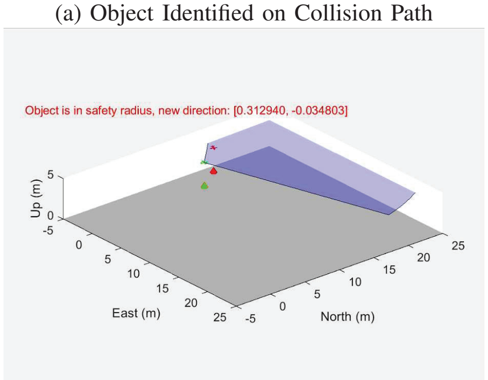
The avoidance sensor and algorithm were simulated in MATLAB. A Python-based Euclidean velocity vector algorithm smoothed the transition between MATLAB and the SITL drone. With MATLAB, Gazebo, and QGC linked, the MATLAB simulation ran the avoidance scenario, providing vectors to the Gazebo environment for real-time monitoring



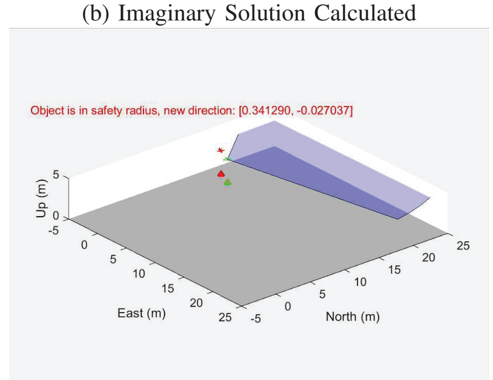
(a) Object Identified on Collision Path



(b) Imaginary Solution Calculated



(c) Object in Safety Radius



(d) Successful Avoidance

Fig. 2: VO/ APF Avoidance Simulation - Green (UAV), Red (Object)

(a) Initial Reward Structure

(b) Final Reward Structure

Fig. 3: Model Final Score Performance Comparison

and adjustment of the SITL drone's movements. Full deviations from the flight plan could then be observed and modified as needed.

The QGC simulation visualized constraints from the weather station and flight path redirection by the DQN. Waypoint and constraint data were sent to the DQN for analysis and redirection, resulting in a new route in the WGS-84 coordinate system, which was applied to the simulated drone.

In Fig. 4a, the initial flight plan is shown in orange, with weather constraints visualized as red polygons along the projected course. The DQN then calculates a new set of waypoints to avoid these constraints, plotting the revised route in white. In Fig. 4b, additional constraints from the wind sensor and a simulated location along the updated route are incorporated. The DQN receives this real-time information and further updates the route, as seen in Fig. 4c. Finally, in Fig. 4d additional constraints are generated by the weather sensor to simulate increasing wind volatility.

IV. Conclusion

The autonomous drone landscape is evolving rapidly, driving the integration of machine learning technologies for enhanced real-time operational capabilities. To ensure their safe operation within complex and dynamic environments, this paper has proposed an automated system that prioritizes operational capability, decision-making, and adaptability. By emphasizing real-time object detection and avoidance using millimeter wave radar rather than traditional LIDAR systems, UAVs can effectively identify and circumvent obstacles while maintaining lightweight, low-power requirements.

(a) Initial Redirection (White) from Flight Plan (Orange)

(b) Constraints Generated from Wind Sensor

(c) Redirection (White) from Flight Plan (Orange)

(d) Additional Constraints Generated from Wind Sensor Data

Fig. 4: Wind Constraints Creation and DQN Flight Redirection

In addition, recognizing a UAV's susceptibility to adverse weather conditions, this paper presented an on-drone system for weather and location data collection that transmits information to a ground station. Here, real-time constraint boundaries are established to mitigate risks effectively. This data collection and transmission process plays a critical role in enhancing navigational accuracy and contributing to broader aerial traffic management.

Furthermore, this paper has highlighted the importance of proactive navigation and long-range planning through in-flight data analysis. By proposing the use of a Deep Q-Network (DQN) for intelligent navigation and air corridor traffic management, the system can anticipate and adapt to changing conditions, ensuring that UAVs can navigate efficiently and safely. This proactive approach enhances operational efficiency and reliability, providing a promising framework for the future of autonomous UAV logistics and transportation.

Future work will focus on expanding the DQN model to support multiple UAV routes and linking additional weather sensors to refine the constrained airspace areas. Additionally, real-world tests are planned to validate its effectiveness. These future developments aim to create a more robust and scalable solution for UAV traffic management, ultimately leading to safer and more efficient airspace utilization. As the technology matures, integrating more sophisticated machine learning algorithms and advanced sensor networks will be crucial in addressing emerging challenges and further enhancing the operational capabilities of autonomous UAVs.

References

[1] Q. Liang, Z. Wang, Y. Yin, W. Xiong, J. Zhang, and Z. Yang, "Autonomous aerial obstacle avoidance using LiDAR sensor fusion," *PLoS One*, vol. 18, no. 6, p. e0287177, Jun. 2023.

[2] K.-S. Hwang, J.-L. Lin, and K.-H. Yeh, "Learning to adjust and refine gait patterns for a biped robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 12, pp. 1481–1490, 2015.

[3] N. Kodama, T. Harada, and K. Miyazaki, "Traffic signal control system using deep reinforcement learning with emphasis on reinforcing successful experiences," *IEEE Access*, vol. 10, pp. 128 943–128 950, 2022.

[4] Texas Instruments, *xWR1843 Evaluation Module (xWR1843BOOST) Single-Chip mmWave Sensing Solution User's Guide*, Texas Instruments Incorporated, May 2020, Revised edition of SPRUIM4B – December 2018. [Online]. Available: www.ti.com

[5] A. Bochkovskiy, C.-Y. Wang, and H.-y. Liao, "Yolov4: Optimal speed and accuracy of object detection," 04 2020.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *European conference on computer vision*, pp. 740–755, 2014, referenced for Tiny COCO dataset subset. [Online]. Available: https://cocodataset.org/

[7] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[8] R. Szczepanski, T. Tarczewski, and K. Erwinski, "Energy efficient local path planning algorithm based on predictive artificial potential field," *IEEE Access*, vol. 10, pp. 39 729–39 742, 2022. [Online]. Available: https://doi.org/10.1109%2Faccess.2022.3166632

[9] Calypso Instruments, *ULTRA-LOW-POWER ULTRASONIC WIND SENSOR (ULP STD)*, C/Alfonso Solans Serrano 20, local 12, Zaragoza, Spain 50014, May 2023, english version 1.0. [Online]. Available: http://calypsoinstruments.com