Title

To Develop Path Planning and Obstacle Avoidance for a Robotic Arm Mounted on a Quadrotor UAV

by B.Yashwanthkumar

Abstract

The study is aimed at improving the autonomy of unmanned aerial vehicles by implementing a robotic arm with advanced path planning and obstacle avoidance tactics. The objective of this project is to build a UAV system that will navigate through complex environments with both static and dynamic obstacles and perform pick-and-place functions. An advanced version of the a-star algorithm is used to find the optimal path and move efficiently. The simulation environment is built with VPython, NumPy, and Matplotlib and virtualizes a real-world environment by using an occupancy grid to detect obstacles and a dynamic grid to track moving obstacles. A three-fingered robotic arm perfectly grasped and released the object, showing precise object manipulation. The UAV moved between waypoints, picked up an object, and moved it to a drop off location avoiding obstacles. The results of the simulation demonstrated the effectiveness of the proposed method by showing improved navigation and successful task execution. The integrated system can be used for automated delivery, disaster relief, industrial automation, and environmental tasks. The paper contributes to the progress of autonomous UAV technology because of the use of a robotic arm, advanced path planning and navigation, and obstacle avoidance tactics. The proposed method makes UAV and aerial robots more adaptive and intelligent.

Introduction

UAV have started to be more commonly used in carrying out autonomous tasks, but the ability to interact with the environment is still limited. Therefore, this project is developing a quadrotor UAV with a robotic arm for navigation in complex environments and object manipulation. The A* algorithm is being improved for path planning, and a 3D simulation of the new UAV model is being created with object manipulation and obstacle avoidance to display the capabilities of the UAV to perform tasks efficiently in dynamic environments with minimal human input.

Problem Statement

To develop an autonomous system for a quadrotor UAV with a mounted robotic arm that can navigate through environments with both static and dynamic obstacles while performing pick-and-place operations, using advanced path planning and obstacle avoidance algorithms.

Objective

Project objective is to create autonomous quadrotor UAV with robotic arm for operation in challenging environments, specifically for pick and place, and thus, fill the niche for intelligent aerial system used in process automation, disaster management, and industry. This involves enhancing A* algorithm for path planning, implementing dynamic obstacle avoidance, and creating three-fingered robotic gripper for object manipulation. A 3D simulation is performed to analyze UAV precision, collision avoidance, and overall operability, and shows that its implementation can boost the operational safety, efficiency, and flexibility.

Relevance

This project has a high degree of relevance, as it significantly raises the autonomy of UAVs by integrating advanced path planning and robotic manipulation, which allows such devices to carry out the operator's tasks with high accuracy in dynamic environments. The project contributes to solving the problem of avoiding obstacles and handling objects, which will increase the efficiency of the use of UAVs for the automatic performance of tasks such as the delivery of goods, emergency response, and industrial automation. Thus, the project's ability to move and interact with objects increases the practical application of the use of UAVs in real conditions.

Literature Review

The field of UAV path planning and obstacle avoidance has gained significant research attention, leading to various methodologies that enhance autonomous navigation. Among these, A and artificial potential field (APF) algorithms* have been widely studied for efficient obstacle avoidance in dynamic environments. Tang et al. (2024) proposed an improved A algorithm* combined with the artificial potential field (APF) method, allowing a 6-DOF robotic arm to navigate around obstacles efficiently while reducing computational overhead [1]. Similarly, Farid et al. (2022) modified the A approach* to optimize UAV motion planning in 3D cluttered environments, improving the system’s efficiency in path selection [2]. These studies highlight the effectiveness of heuristic search algorithms in real-time applications but also reveal limitations in handling unpredictable obstacles.Motion planning for aerial pick-and-place tasks has been another area of interest. Cao et al. (2025) explored geometric feasibility constraints to enhance UAV movement while maintaining precision in grasping and placing objects [3]. The application of artificial intelligence in UAV navigation has also been investigated through cooperative UAV systems. Rao et al. (2023) integrated A and artificial potential field* techniques to optimize dual-UAV cooperative transport, ensuring minimal collision risks and smoother trajectory planning [4]. Gomathi and Rajathi (2023) further emphasized the adaptive nature of path planning in unknown environments, where UAVs must continuously adjust their trajectories based on real-time sensor data [5].Recent advancements have explored reinforcement learning (RL) and deep learning for UAV trajectory optimization. Wang et al. (2020) applied deep reinforcement learning to UAV navigation, demonstrating improvements in obstacle avoidance and trajectory adaptation in dynamic environments [9]. Luo et al. (2018) proposed a Deep-Sarsa-based multi-UAV planning framework, where UAVs learn from previous navigation patterns, improving their ability to avoid obstacles autonomously [11]. These approaches showcase the potential of machine learning models in refining UAV path planning strategies. Optimization techniques have also played a crucial role in UAV path planning. Hashemi and Heidari (2020) introduced an optimal control approach for UAV trajectory planning with a suspended payload, ensuring minimal oscillations and efficient movement [6]. Additionally, Wang (2023) leveraged ant colony optimization to enhance multi-source information fusion technology, refining the UAV's ability to navigate through complex terrains [7]. These studies contribute to the growing body of research aimed at reducing energy consumption, improving computational efficiency, and ensuring real-time decision-making.The foundation of motion planning and obstacle avoidance is further supported by early studies in robotic path planning. Minguez et al. (2008) provided a comprehensive framework for motion planning, detailing essential concepts that remain relevant in today's UAV navigation systems [8]. Similarly, Trung Tat Pham and Defigueiredo (1991) pioneered adaptive robot path planning in dynamic environments, laying the groundwork for modern UAV applications [12].

Research Gap

While UAV path planning has progressed significantly, there are still unresolved challenges that limit its real-world effectiveness. Traditional algorithms like A and artificial potential field (APF)* have been extensively used, but they often struggle in dynamic environments where obstacles move unpredictably (Tang et al., 2024; Farid et al., 2022) [1,2]. These methods rely on predefined heuristics and lack the ability to adapt efficiently in real-time. Recent advancements in deep reinforcement learning (DRL) have improved UAV navigation capabilities, yet high computational costs and slow decision-making remain key obstacles when applied to complex environments (Wang et al., 2020; Luo et al., 2018) [9,11].

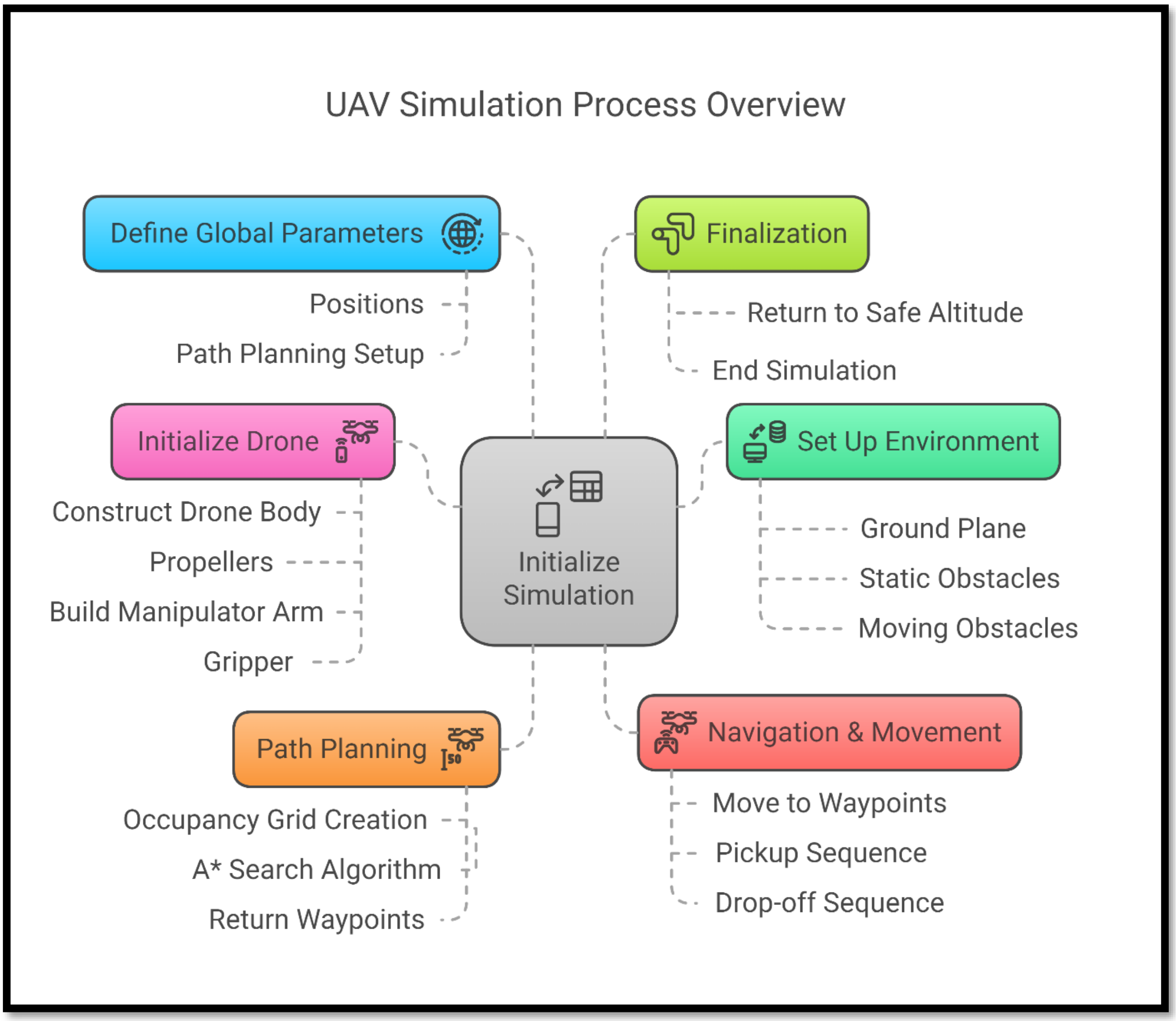
Additionally, most existing studies have been limited to simulations, with fewer real-world applications that consider factors like environmental uncertainties, varying wind conditions, and human interactions (Hashemi & Heidari, 2020; Wang, 2023) [6,7]. Another critical challenge is multi-UAV coordination, as effective communication and collision avoidance strategies for swarm UAVs remain underexplored (Cao et al., 2025) [3]. Although hybrid models that integrate heuristic optimization, deep learning, and bio-inspired algorithms have been proposed, their practical feasibility and efficiency in real-time UAV deployment require further investigation. Addressing these gaps will require developing lightweight DRL models, optimizing real-time sensory processing, and conducting field tests in diverse and unstructured environments to ensure robust performance in real-world applications.

Proposed Approach

The project differs from traditional methods that are based on static obstacles. The project uses the real-time adjustments of the navigation. These adjustments depend on the data about the environment. The A* algorithm was developed specifically for this project. It selects the path and minimizes the deviations from the target. Also, the UAV is equipped with a robotic gripper, which has three fingers. This allows to manipulate the object. The system works in a realistic 3D simulation environment. The environment is built with VPython, NumPy, and Matplotlib. Different stationary and moving obstacles are used to test its flexibility. By combining the navigation, obstacle avoidance, and object manipulation in one autonomous system, the project makes the UAV more versatile. The UAV can be used in many areas, such as emergency response, automatic delivery, and industrial robotics.

Methodology

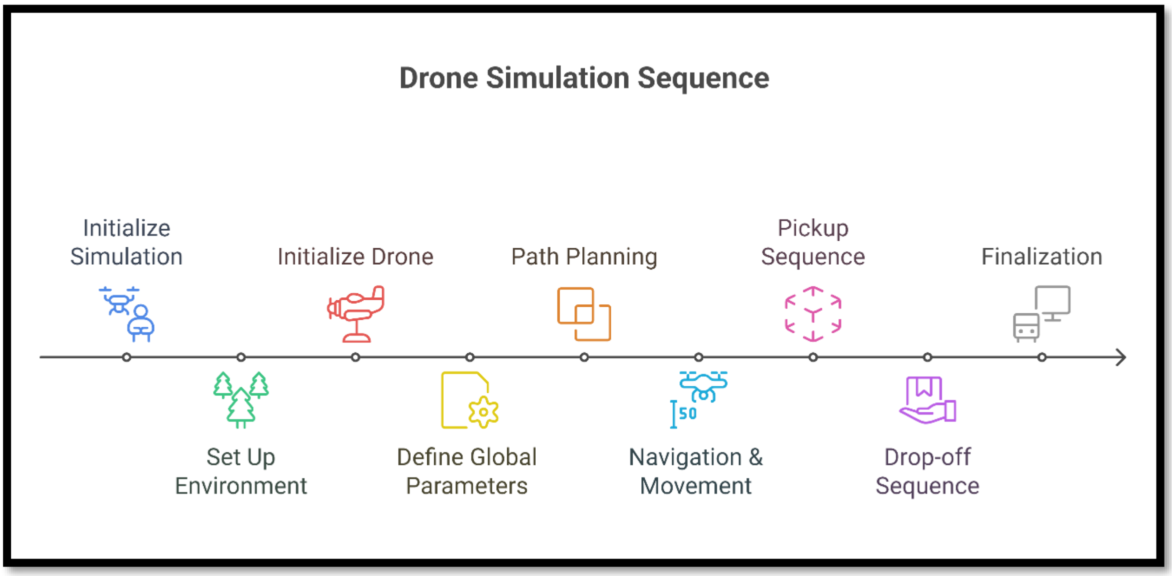
The robotic system in the project comprises a quadrotor UAV with a robotic arm in the center. The UAV is designed for autonomous navigation and manipulation of objects. First of all, the quadrotor UAV has four propellers that make the flight stable. The robotic arm with a three-fingered gripper provides a secure grip and accurate handling of the object. The A* algorithm was improved to calculate the optimal path with a static and dynamic obstacle. Real-time tracking allows adaptive movement of the robotic arm to avoid obstacles. The ability of the robotic arm to extend and retract allows efficient pick and place operations. The integration of UAV and robotic arm improves the overall capabilities of UAVs for industrial automation, delivery, and other applications.



Algorithm Steps

The Structured development of autonomous quadrotor UAV with robotic arm is proposed for efficient navigation, obstacle avoidance, and object manipulation.

1. **Simulation Environment Setup:** The 3D simulation is built using VPython NumPy and Matplotlib which involves the UAV model and obstacles and targets for testing.
2. **UAV and Robotic Arm Modeling:** The quadrotor UAV is designed with four propellers and a robotic gripper.
3. **Path Planning:** The optimized A-star algorithm computes the shortest path, avoiding obstacles ensuring safety.
4. **Real-Time Obstacle Avoidance:** The UAV identifies and maintains obstacle awareness and dynamically alters its course to prevent obstacles collision.
5. **Pick-and-Place Execution:** Unmanned Aerial Vehicle approaches the selected object, grasps, translates, puts the object and leaves with the coordinated movement of the arm.
6. **Performance Evaluation:** Navigation accuracy, obstacle avoidance efficiency, and task completion time are analyzed to validate system effectiveness.



Pseudocode

```
BEGIN Drone_Navigation_Simulation
  // Initialize Simulation Environment
  CREATE 3D simulation environment with VPython
  SET scene parameters (title, dimensions, background)
  CREATE ground surface (green box)
  // Initialize Static Obstacles
  FOR each defined obstacle position:
    CREATE box obstacle at position with specified size and color
  // Initialize Moving Obstacles
  CREATE moving obstacles as spheres with different colors
  STORE moving obstacles in a list
  // Initialize UAV Components
  CREATE drone_body as a box (light brown color)
  DEFINE propeller_offsets relative to drone_body
  FOR each propeller_offset:
    CREATE fan propeller using create_fan_propeller(drone_body.position + offset)
  CREATE manipulator arm as cylinder and arm_joint as sphere
  FOR each angle in [0, 120, 240]:
    CREATE gripper finger using create_finger(angle)
  GROUP gripper fingers into gripper and update their positions
  // Initialize Object to Manipulate
  CREATE object (blue sphere) for pick-and-place operation
  // Initialize Tracking Variables
  INITIALIZE uav_path as empty list
  INITIALIZE moving_obstacle_paths as list of empty lists
  SET grasped flag to FALSE
  INITIALIZE time counter t to 0
  // Define Obstacle Update Function
  FUNCTION update_moving_obstacles(dt):
    INCREMENT t by dt
```

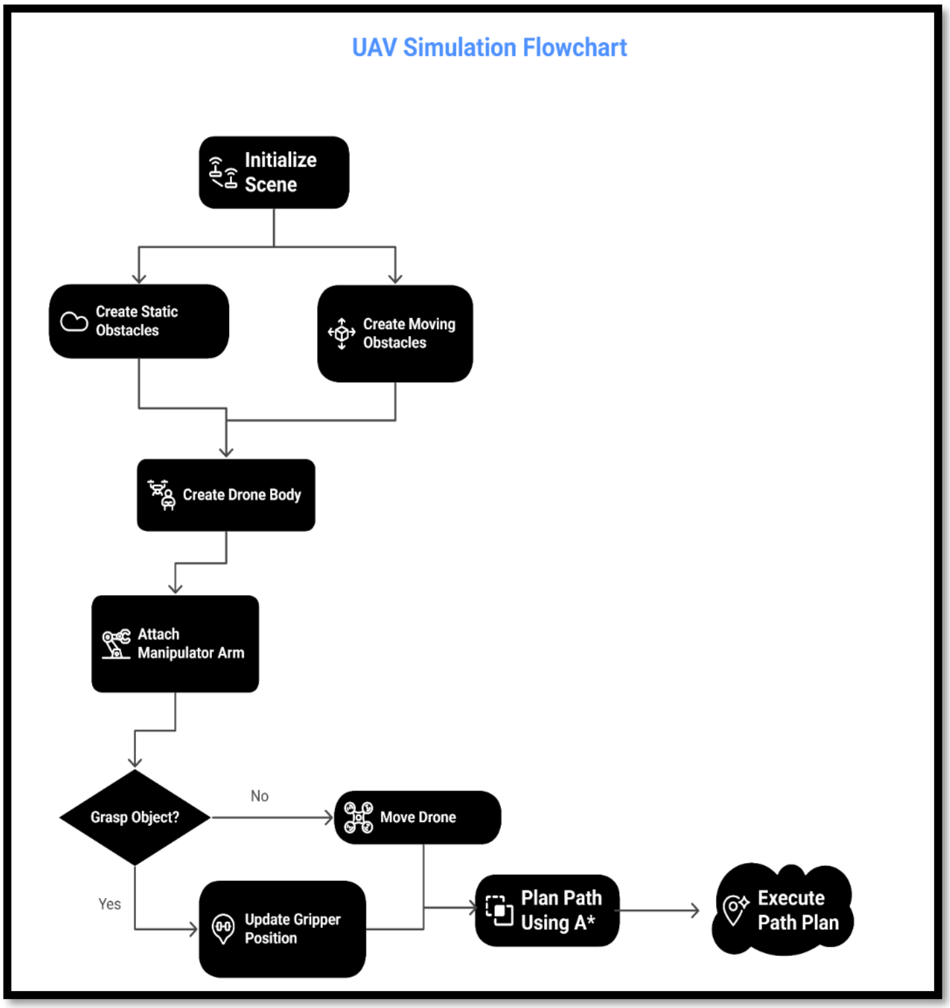
```
  FOR each moving obstacle:
    UPDATE obstacle position using sine/cosine functions based on t
    RECORD obstacle position into corresponding moving_obstacle_paths
// Movement Functions
FUNCTION move_to(target, duration):
  CALCULATE number of steps based on duration
  FOR each step:
    INTERPOLATE drone_body.position from current to target
    UPDATE drone_body position and record new position in uav_path
    FOR each propeller:
      UPDATE propeller.position relative to drone_body
      ROTATE propeller to simulate fan motion
    UPDATE arm.position and arm_joint.position based on drone_body
    CALL update_gripper() to reposition gripper fingers
    IF grasped is TRUE THEN:
      SET object.position to arm_joint.position
    CALL update_moving_obstacles(dt)
FUNCTION extend_arm(target_length, duration):
  CALCULATE number of steps from duration
  FOR each step:
    INTERPOLATE arm.axis length from current to target_length
    UPDATE arm.axis and arm_joint.position
    CALL update_gripper()
    IF grasped is TRUE THEN:
      SET object.position to arm_joint.position
    CALL update_moving_obstacles(dt)
FUNCTION grasp():
  SET grasped flag to TRUE
  ATTACH object to arm_joint.position
FUNCTION release(target_pos):
  SET grasped flag to FALSE
```

```
    SET object.position to target_pos
// A* Path Planning Algorithm
FUNCTION plan_path_Astar(start, goal):
    INITIALIZE grid parameters: resolution, margin, x_min, x_max, z_min, z_max
    FUNCTION coord_to_grid(pos):
        CONVERT world coordinate pos to grid indices
    FUNCTION grid_to_coord(cell):
        CONVERT grid indices cell to world coordinate (cell center)
    CREATE occupancy grid by marking cells occupied by static obstacles (with margin)
    CONVERT start and goal to grid cells (start_cell, goal_cell)
    INITIALIZE open_set (priority queue), g_score, f_score, and closed_set
    WHILE open_set is NOT EMPTY:
        REMOVE cell with lowest f_score as current
        IF current equals goal_cell THEN:
            RECONSTRUCT path from goal to start using came_from
            CONVERT grid cells in path to world coordinates
            RETURN path
        ADD current to closed_set
        FOR each neighbor in 8 directions:
            IF neighbor is within grid bounds AND NOT occupied AND NOT in closed_set THEN:
                COMPUTE tentative_g score from current to neighbor
                IF tentative_g < g_score for neighbor THEN:
                    UPDATE came_from, g_score, and f_score for neighbor
                    ADD neighbor to open_set
    RETURN direct path [goal] as fallback
// Navigation Sequence
SET start_pos, pickup_point, dropoff_point
CALCULATE above_pickup = pickup_point + vertical offset
CALCULATE above_dropoff = dropoff_point + vertical offse
path1 ← plan_path_Astar(start_pos, above_pickup)
FOR each waypoint in path1:
```

```
    CALL move_to(waypoint, specified duration)
  CALL move_to(pickup_point, specified duration)
  CALL extend_arm(lower arm to grasp position, specified duration)
  CALL grasp()
  CALL extend_arm(retract arm to safe position, specified duration)
  CALL move_to(above_pickup, specified duration)
  path2 ← plan_path_Astar(above_pickup, above_dropoff)
  FOR each waypoint in path2:
    CALL move_to(waypoint, specified duration)
  CALL move_to(dropoff_point, specified duration)
  CALL extend_arm(lower arm to drop position, specified duration)
  CALL release(dropoff_point)
  CALL extend_arm(retract arm to safe position, specified duration)
  CALL move_to(above_dropoff, specified duration)
  PRINT "Simulation complete."
  // Visualization and Animation
  INITIALIZE Matplotlib 3D figure
  PLOT UAV path and moving obstacle paths in 3D space
  SET animation functions for dynamic visualization
  EXECUTE simulation animation using Matplotlib
END Drone_Navigation_Simulation
```

Simulation Initialization: The UAV, robotic arm, obstacles, and environment are created using VPython.
Path Planning (A*Algorithm):The UAV determines an optimized path to avoid static obstacles.
Real-Time Obstacle Avoidance: Moving obstacles follow sinusoidal trajectories, requiring continuous tracking.
Pick-and-Place Operations: Robotic arm extends, grasps, transports, and releases object.
Navigation Sequence: The UAV follows a structured route with waypoints, dynamically adjusting to obstacles.
Visualization & Animation: The system records movement paths and visualizes them in 3D using Matplotlib.

Flowchart



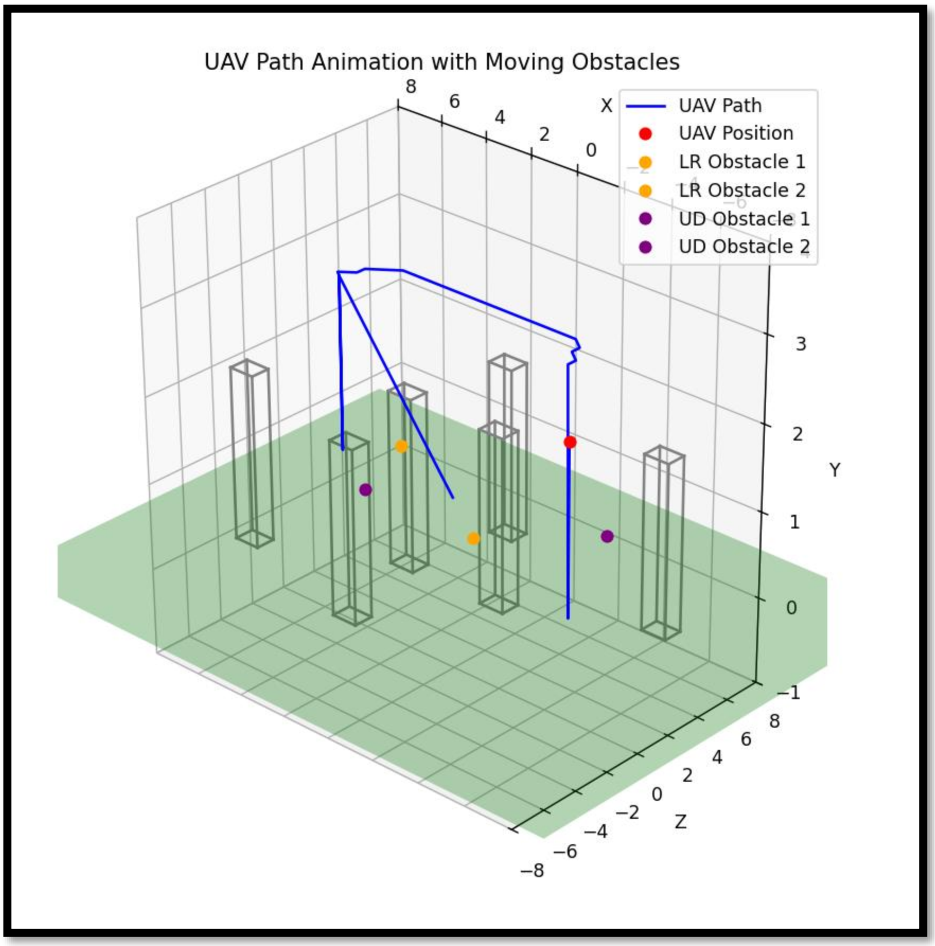
Research Design

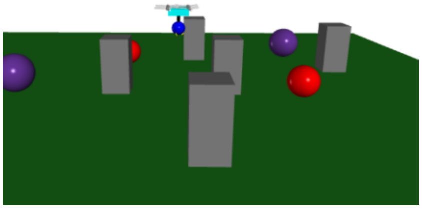
This Project followed mixed-methods research design blending UAV performance quantitative analysis and obstacle avoidance efficiency qualitative evaluation. Quantitative method observes navigation accuracy, path deviation and task completion time. Qualitative evaluation method assesses adaptability of UAV in dynamic environments. The study integrates simulation-based testing and real-time tracking to ensure a thorough evaluation UAV autonomous navigation and pick and place abilities.

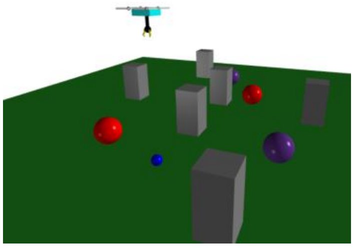
Results and Discussion

Software Tools Used to Implement the Project

1. **Python** is the primary programming language; it is used for UAV simulation and path planning.
2. **VPython** for three-dimensional (3D) visualizing of the UAV, robotic arm, and the environment.
3. **NumPy** offers support for mathematical and numerical operations.
4. **Matplotlib** is used to visualize and animate the UAV's path and the moving obstacles.
5. **Mplot3D (Matplotlib Toolkit)** is used for 3D plotting of the UAV trajectories and obstacle.
6. **Heapq module** has been used. It provides a min-heap implementation to be used as a priority queue for A*.
7. **Math module** performs mathematical calculations (trigonometric functions) for motion updates.

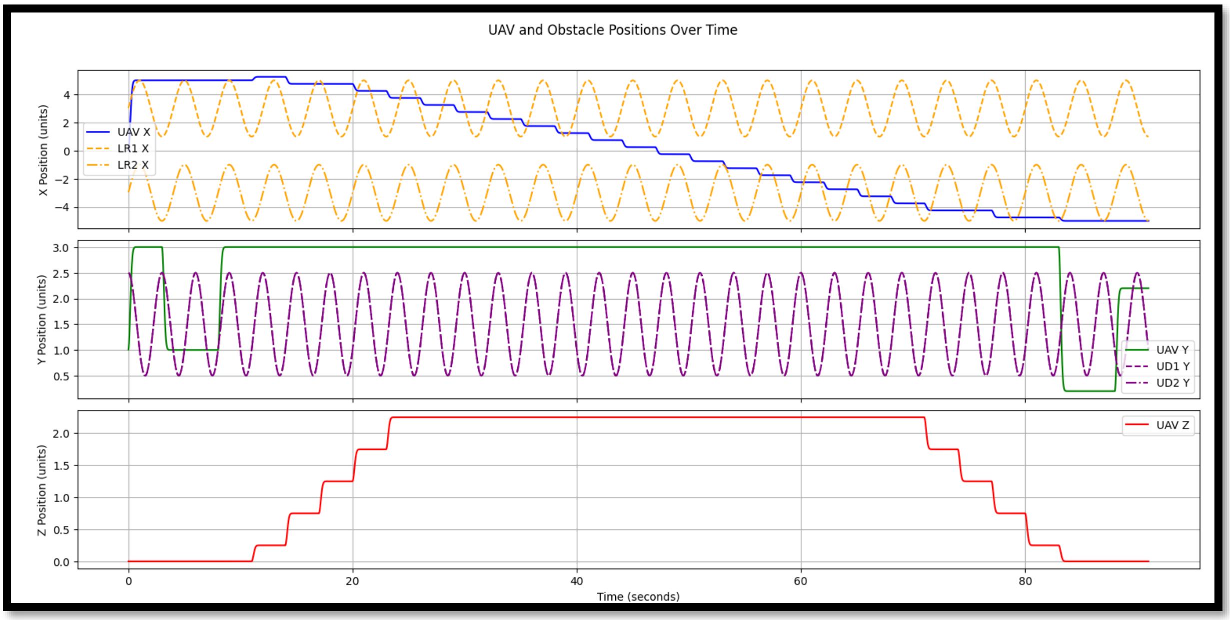






Initial Position **Middle Position** **Final Position**

Graph



The simulation effectively verified the Unmanned Aerial Vehicle's maneuverability in complex environments and obstacle avoidance against static and dynamic obstacles improved by the A* algorithm. The UAV followed the computed path precisely and accurately, and whenever the moving obstacles encountered, the UAV adjusted its route accordingly. The robotic arm's Pick-and-Place operation was completed accurately to ensure stable grasping, transportation, and object release at the specified locations.

The Plotted data shows the UAV relocating steadily in the X-direction. It modifies its trajectory dynamically with visible traces of obstacles. The observed smooth path allows suggesting the obstacle avoidance system performs properly. The UAV manages to avoid obstacles and not to collide with them as well. The observed graph that shows the UAV's altitude and the discovered pattern gives an idea of a controlled vertical response. The UAV adapts to a three-dimensional environment.

The results confirm that the system can effectively be used in real-world scenarios such as autonomous delivery, surveillance, and search-and-rescue operations. From the graph, it is clear that the obstacles exhibit oscillatory motion in both X and Y directions, providing a dynamic challenge for the UAV. Nonetheless, the UAV successfully navigates through the environment without abrupt deviations, signifying real-time decision-making and smooth path execution. The gradual descent in the X-position further proves the UAV's ability to maintain a planned trajectory while actively responding to obstacle movements.

These findings highlight the effectiveness of the navigation algorithm in real-time obstacle avoidance, suggesting potential improvements in computational efficiency and adaptability for more complex terrains.

Conclusion and Recommendations

Summary of Key Findings

The successful demonstration of the UAV's capability to navigate dynamic environments and effectively avoid obstacles was accomplished. The results supported the claim that the obstacle avoidance algorithm can provide smooth modifications of the trajectory in real time and, thus, collision-free movement. The UAV kept stable altitude modifications while responding to the obstacles in the X and Y directions, which proves the efficiency of the system. The obtained results can be used to implement the proposed approach in the real-life applications, such as autonomous surveillance, the logistics of goods delivery, and emergency response.

Practical or Theoretical Contributions

The contribution of this research is towards the advancement in autonomous UAV navigation by presenting a highly effective real-time decision-making model. In practice, the system can significantly enhance UAV-based applications with dynamic obstacle handling requirements. In theory, the findings advance the understanding of trajectory planning algorithms and multi-dimensional obstacle avoidance strategies. This understanding can be used for further development of robotic path planning as well as adaptive AI-driven navigation systems.

Limitations and Suggestions for Future Research

UAV navigated dynamic obstacles successfully, but responses were based on pre-defined movement patterns. This issue could be addressed in the future by incorporating machine learning-based predictive models, especially reinforcement learning to enhance adaptability. Reinforcement learning would enable UAV to learn from the environment and dynamically adjust its trajectory, improving decision-making in unpredictable scenarios. Testing in real-world outdoor environments, where obstacles like humans and vehicles move unpredictably would further validate the reliability of the system. Enhancing computational efficiency and reducing response time would also be necessary for seamless real-time operation under complex and dynamic conditions.

References:

[1] Tang X, Zhou H, Xu T. Obstacle avoidance path planning of 6-DOF robotic arm based on improved A* algorithm and artificial potential field method. *Robotica*. 2024;42(2):457-481. doi:10.1017/S0263574723001546

[2] Farid, G.; Cocuzza, S.; Younas, T.; Razzaqi, A.A.; Wattoo, W.A.; Cannella, F.; Mo, H. Modified A-Star (A*) Approach to Plan the Motion of a Quadrotor UAV in Three-Dimensional Obstacle-Cluttered Environment. *Appl. Sci.* **2022**, *12*, 5791.

[3] H. Cao, J. Shen, C. Liu, B. Zhu and S. Zhao, "Motion Planning for Aerial Pick-and-Place With Geometric Feasibility Constraints," in IEEE Transactions on Automation Science and Engineering, vol. 22, pp. 2577-2594, 2025, doi: 10.1109/TASE.2024.3382296.

[4] Jinjun Rao, Chaoyu Xiang, Jinyao Xi, Jinbo Chen, Jingtao Lei, Wojciech Giernacki, Mei Liu, Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm, Knowledge-Based Systems,Volume 277,2023,110797,ISSN 0950-7051,doi.org/10.1016/j.knosys.2023.110797.0

[5] Gomathi N, Rajathi K. Adaptive path planning for unknown environment monitoring. Journal of Ambient Intelligence and Smart Environments. 2023;15(4):287-314. doi:10.3233/AIS-220175.

[6] Hashemi, D., Heidari, H. Trajectory Planning of Quadrotor UAV with Maximum Payload and Minimum Oscillation of Suspended Load Using Optimal Control. *J Intell Robot Syst* **100**, 1369–1381 (2020). doi:10.1007/s10846-020-01166-4

[7] Wang, M. (2023). Research on Quadrotor UAV Path Planning Optimization Based on Multi-source Information Fusion Technology of Ant Colony Optimization Algorithm. In: Liang, Q., Wang, W., Liu, X., Na, Z., Zhang, B. (eds) Communications, Signal Processing, and Systems. CSPS 2022. Lecture Notes in Electrical Engineering, vol 872. Springer, Singapore,doi:10.1007/978-981-99-2653-4_20

[8] Minguez, J., Lamiraux, F., Laumond, JP. (2008). Motion Planning and Obstacle Avoidance. In: Siciliano, B., Khatib, O. (eds) Springer Handbook of Robotics. Springer, Berlin, Heidelberg, doi:10.1007/978-3-540-30301-5_36

[9] Wang, G., Zheng, X., Zhao, H., Zhao, Q., Zhang, C., Zhang, B. (2020). Unmanned Aerial Vehicles Path Planning Based on Deep Reinforcement Learning. In: Liu, Y., Wang, L., Zhao, L., Yu, Z. (eds) Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery. ICNC-FSKD 2019. Advances in Intelligent Systems and Computing, vol 1074. Springer, Cham, doi:10.1007/978-3-030-32456-8_9

[10] L. Chang, L. Shan, J. Li and Y. Dai, "The Path Planning of Mobile Robots Based on an Improved A* Algorithm," 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 2019, pp. 257-262, doi: 10.1109/ICNSC.2019.8743249. keywords: {Smoothing methods;Path planning;Turning;Robot kinematics;Mobile robots;Automation;mobile robot;path planning;map compression;smooth A* algorithm},

[11] Luo, W., Tang, Q., Fu, C., Eberhard, P. (2018). Deep-Sarsa Based Multi-UAV Path Planning and Obstacle Avoidance in a Dynamic Environment. In: Tan, Y., Shi, Y., Tang, Q. (eds) Advances in Swarm Intelligence. ICSI 2018. Lecture Notes in Computer Science(), vol 10942. Springer, Cham,doi.10.1007/978-3-319-93818-9_10

[12] Trung Tat Pham and Rui J. Defigueiredo. 1991. Adaptive robot path planning with obstacle avoidance in a dynamic environment. Ph.D. Dissertation. Rice University, USA. Order Number: AAI9136060.

[13] Ait Saadi, A., Soukane, A., Meraihi, Y. *et al.* UAV Path Planning Using Optimization Approaches: A Survey. *Arch Computat Methods Eng* **29**, 4233–4284 (2022). doi:10.1007/s11831-022-09742-7

[14] L. Chang, L. Shan, J. Li and Y. Dai, "The Path Planning of Mobile Robots Based on an Improved A* Algorithm," 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 2019, pp. 257-262, doi: 10.1109/ICNSC.2019.8743249. keywords: {Smoothing methods;Path planning;Turning;Robot kinematics;Mobile robots;Automation;mobile robot;path planning;map compression;smooth A* algorithm},

[15] Xiangyu Wang, Zuoshi Liu, Jiahu Liu, "Mobile robot path planning based on an improved A* algorithm," Proc. SPIE 12604, International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2022), 1260449 (23 May 2023); doi:10.1117/12.2674526