AUTONOMOUS ROUTE PLANNING IN DYNAMIC ENVIRONMENTS (DYNARL)
HAVELSAN A.S.

PROJECT TEAM
Berkay Altıntaş
Berkehan Ercan
Ahmet Arda Kocabağ
Muhammet Bahadır Mutlu
Emirhan Tekez

COMPANY MENTORS
Dr. Çağlar Akman
Ahmet Gökhan Göze
Hasan Furkan Oztürk
Esat Serhat Sucu
ACADEMIC MENTOR
Prof. Arif Bülent Özgüler
TEACHING ASSISTANT
Mehmet Efe Lorasdağı

Abstract. This project presents an autonomous unmanned aerial vehicle (UAV) system for navigating dynamic indoor environments using real-time perception and decision-making. The system integrates simultaneous localization and mapping (SLAM), YOLO-based object detection, and reinforcement learning (RL) for collision-free path planning in indoor environments. The UAV is equipped with a 3D LIDAR, depth camera, and IMU, and builds a 3D reconstruction of its unknown environment during its route to the specified target location. The RL policy, which was trained using 700 parallel agents in NVIDIA Isaac Sim, generates velocity commands in the drone's local frame. The model was tested in Gazebo, achieving an 87% collision-free success rate. All functional and most non-functional requirements were met. The project could further be extended to multi-agent systems swarm systems and embedded hardware deployment.

PROJECT DESCRIPTION

The project *Autonomous Route Planning in Dynamic Environments* aims to develop an unmanned aerial vehicle (UAV) system capable of autonomously navigating in indoor spaces by avoiding both static and dynamic obstacles. This work addresses the real-world challenge of enabling UAVs to operate independently in real time, particularly in situations where human control is infeasible or unsafe—such as post-disaster areas, infrastructure inspection sites, or urban military environments. In response to HAVELSAN's demand for intelligent UAV platforms, the team proposes a software-simulated system capable of real-time decision-making, mapping, and route planning, with the potential for future integration into real-world platforms like HAVELSAN's POYRAZ UAV [1].

This project is motivated by the increasing use of UAVs across fields such as logistics, surveillance, search and rescue, and defense. While existing research offers partial solutions—such as *"Visual SLAM in Dynamic Environments Based on Object Detection"* [2], which does not include dynamic obstacle handling; *"Object Detection from the Video Taken by Drone via Convolutional Neural Networks"* [3], which is limited to outdoor environments and a well-known static obstacles such as buildings; and *"Mobile Robot Navigation Using Deep Reinforcement Learning"* [4], which is constrained to 2D navigation without dynamic obstacles—these systems lack the integration of SLAM, object detection, and reinforcement learning in such a way that ensures autonomous platforms can navigate autonomously in indoor spaces with dynamic obstacles, which is the novelty of this project.

To overcome the abovementioned limitations, the proposed solution integrates real-time SLAM, object detection, and reinforcement learning into a single system. A YOLO-based object detection module [5], trained on the MS-COCO2017 dataset [6], handles real-time dynamic obstacle recognition. An RTAB-Map-based SLAM system processes RGB-D and IMU inputs to produce a 3D reconstruction of the UAV's environment. The UAV's control policy is learned through a PPO-based reinforcement learning model, trained using 700 parallel agents in the NVIDIA Isaac [7], and evaluated in Gazebo. The novelty lies in this integration, allowing for real-time adaptability and collision avoidance in complex, dynamic environments. The system was designed in a modular fashion, where each part can be improved independently of the others, allowing it to be used as a testing ground for further algorithm development.

The UAV is equipped with an Intel RealSense D435if depth camera [8], the PX4 flight controller's onboard IMU, and a 3D LiDAR sensor, enabling perception and localization. The system utilizes ROS for communication between the sensors and the onboard computer, RTAB-Map for mapping, YOLOv11n for object detection, and TorchRL for reinforcement learning. In simulation, the model is trained with 700 parallel UAVs in IsaacSim and validated in Gazebo. Performance metrics show the following results: the detection system operates at 60 FPS (far exceeding the 10 FPS requirement), localization accuracy is maintained within ±5 cm at 2 meters, and the crash-free mission success rate is 87%. The average navigation speed of 1 m/s surpasses the minimum threshold of 0.5 m/s, and the SLAM module updates at a rate of 1 Hz, with successful loop closures after a full 360-degree turn.

Big Picture [7] [9] [10].

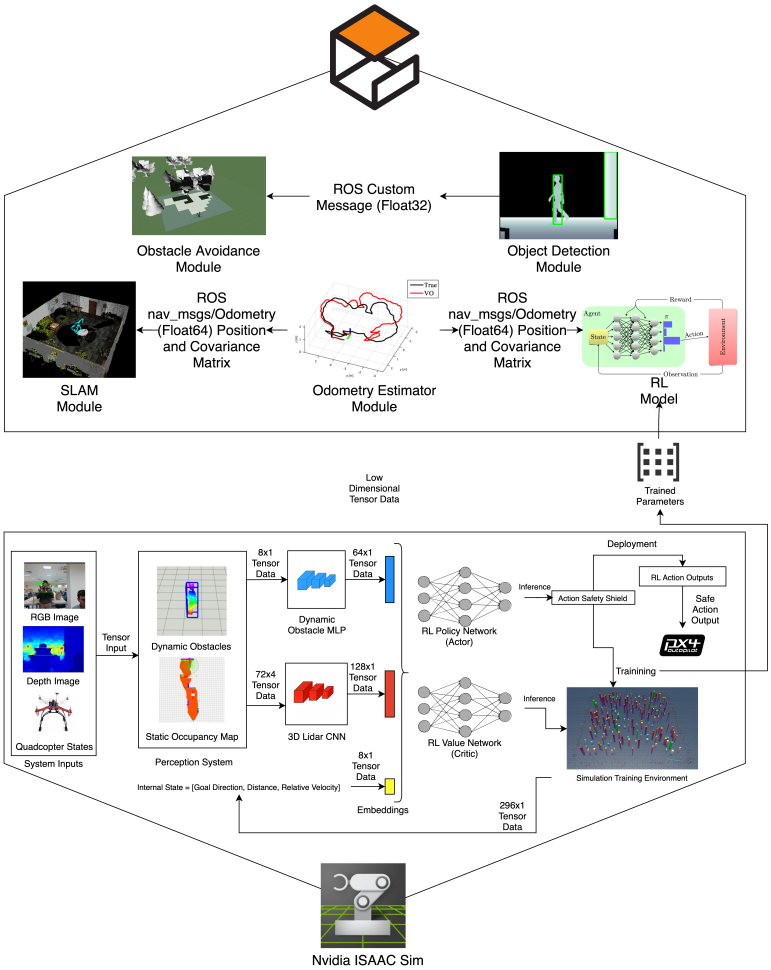


FIGURE 1. Big Picture

The drone's sensor setup includes a depth camera, PX4 IMU, and 3D LiDAR for data collection on both dynamic and static obstacles. The depth camera provides a 1080p 30 FPS RGB feed and a 90 FPS point cloud stream to both the SLAM module and the object detection system. The IMU supplies real-time acceleration, angular velocity, and orientation data via ROS messages of type `sensor_msgs/Imu`.

The SLAM module outputs a 3D reconstruction of the environment as `sensor_msgs/PointCloud2`, supporting localization and mapping. The 3D LiDAR offers a 360-degree scan with a 40-degree vertical FOV, which the RL agent uses for collision avoidance. The RL model processes inputs from LiDAR, depth camera, and IMU. Dynamic obstacle positions are provided by the object detection module via `geometry_msgs/Point`.

Modules including SLAM, odometry, and RL communicate with the PX4 flight controller using the MAVLink protocol to relay position estimates and control commands. The drone's internal state (position and linear velocity) and dynamic obstacle position estimates are used in the decision-making process.

LiDAR data in a 72×4 matrix is passed through a convolutional neural network (CNN) and reduced to a 128×1 vector. This is concatenated with other inputs and processed by a two-layer multilayer perceptron (MLP) with hidden layer sizes of 128 for both layers. The final output is a continuous linear velocity command sent to the flight controller.

This setup enables SLAM, odometry, object detection, and end-to-end reinforcement learning. The RL model is trained in NVIDIA Isaac Sim and later deployed in Gazebo, where it operates using real-time sensor feedback and policy inference.

MILESTONES

I. This milestone covers the implementation of the SLAM module. An RGB-D camera provides 1080p RGB at 30 FPS and point-cloud data at 90 FPS for 3D mapping. The environment is reconstructed by using visual-inertial odometry (VIO) which uses both the depth camera and the IMU. The success criterion for this milestone is achieving a loop closure after a 360° turn for minimal map drift.

II. This milestone covers the implementation of the object detection module. An MS-COCO2017-trained model identifies only the "person" class for dynamic obstacle avoidance. Its accuracy and reliability were tested for safe drone navigation both with the validation set and in Gazebo. The success criterion is to achieve 90% mean average precision (mAP) over the validation set.

III. This milestone covers the training and testing of the RL model in NVIDIA IsaacSim. The success criterion for this milestone is the UAV being able to navigate to its target without crashes more than 85% of the time.

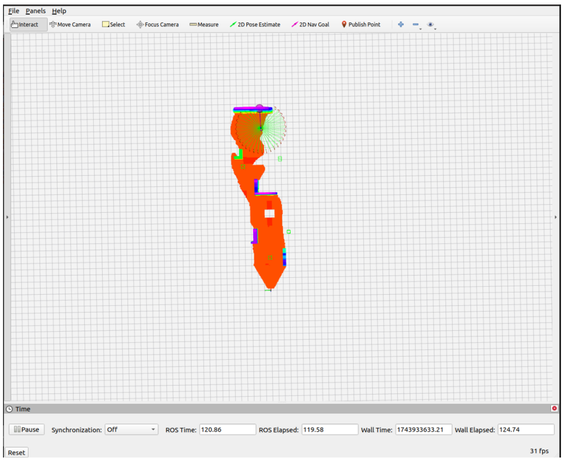
IV. This milestone focuses on integrating the trained RL algorithm into Gazebo. After training, RL communicates with PX4 via ROS messages. Dynamic obstacles with predefined paths were included in the Gazebo, where realistic 3D models were used to ensure the object detection model works properly. The constructed simulation environment includes both static and dynamic obstacles, accurate sensor simulations, and non-idealities, such as sensor noise in both camera and IMU data.

DESIGN DESCRIPTION

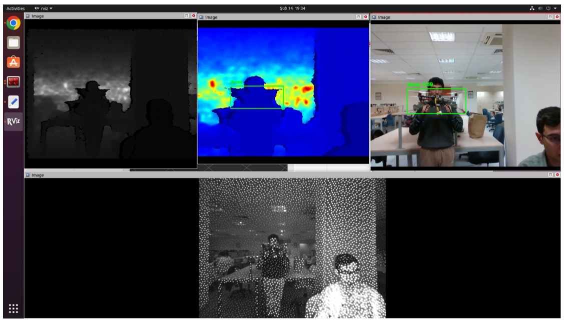
The project was separated into distinct work packages corresponding to each of the major modules mentioned above. Each module was developed and tested independently of one another to ensure modularity. When all work packages concerning SLAM, object detection and RL were completed, they were combined and test alongside one another in Gazebo to ensure the system works properly, with everything attached.

The SLAM module was implemented using RTAB-Map, uses the RGB-D camera alongside the IMU data for 3D mapping and localization. A simulated Intel RealSense D435if camera provided RGB and depth images at 30 and 90 FPS, respectively. IMU data, published as `sensor_msgs/Imu`, was used for localization via visual-inertial odometry. Figure 2a shows the drone's successful 3D reconstruction of its environment, validating the SLAM pipeline.

To detect dynamic obstacles, a YOLOv11n model trained on the MS-COCO dataset was used, with outputs published as `geometry_msgs/Point` messages. This information was used together with the camera's intrinsic parameters and orientation data to estimate the positions of obstacles. Figure 2b shows the detection results.



(A) 3D Reconstruction Output Using RTAB-Map



(B) YOLOv11n Detecting Dynamic Obstacles

FIGURE 2. Visual Outputs of SLAM and Object Detection Modules

The RL agent was trained in NVIDIA Isaac Sim using 3D LiDAR and depth data. A CNN processed the LiDAR input, which was concatenated with other observations, then passed through a two layer MLP. The model's continuous action outputs were sampled from a Beta distribution to ensure smooth, bounded velocities. The reward function focuses on the UAV's alignment with the target direction, current distance from both static and dynamic obstacles and how realizable the trajectory is, by punishing unrealistic turns.

After the training, the model was deployed in Gazebo, where the UAV used ROS messages (`geometry_msgs/TwistStamped`) to navigate toward a target while avoiding dynamic human obstacles. Figure 3 shows the UAV traversing its environment successfully. The final model achieved a success rate of 87%.

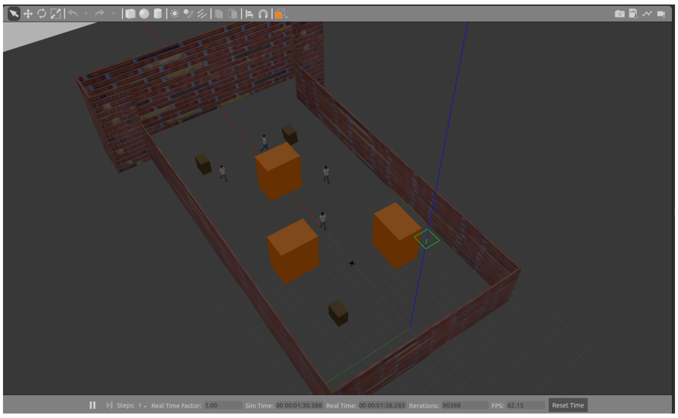


FIGURE 3. UAV Navigating to Target in a Dynamic Gazebo Environment

RESULTS AND PERFORMANCE EVALUATION

The UAV system was tested in a simulated indoor environment which includes static and dynamic obstacles. The reinforcement learning (RL) model successfully guided the drone to its target location while avoiding collisions, achieving an 87% success rate after training (Figure 4).

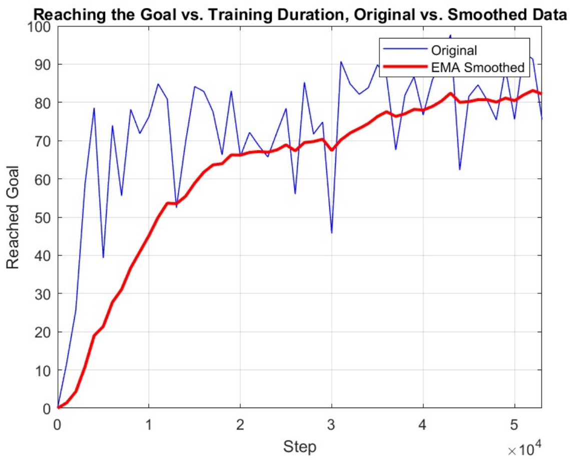


FIGURE 4. RL Model Success Rate During Training

The object detection model, trained on the MS-COCO dataset, reached a 96.4% mean average precision and successfully detected dynamic obstacles in simulation. The UAV maintained an average speed of 1 m/s, exceeding the minimum requirement of 0.5 m/s. All visual and positional data were processed at 60 FPS on a laptop with RTX 3070 GPU.

Compared to works which focus on static 2D environments, this system incorporates SLAM, real-time object detection, and RL-based path planning in a dynamic

3D space. The overall performance meets the project's functional requirements, although real-world deployment and additional safety features remain as future steps.

CONCLUSIONS AND FUTURE DIRECTIONS

In this project, we developed an autonomous navigation system for UAVs capable of operating in dynamic indoor environments by integrating SLAM, object detection, and reinforcement learning. Using simulation platforms such as Gazebo and NVIDIA Isaac Sim, the RL model was trained to navigate complex spaces with both static and dynamic obstacles. The system combined RTAB-Map for environmental mapping, a YOLO-based model for detecting dynamic objects, and a continuous-action RL agent for path planning. The final setup demonstrated real-time autonomous route planning and obstacle avoidance in a realistic simulated environment.

Building on this foundation, future work can focus on transfer of the system from simulation to real-world deployment using embedded hardware platforms. Enhancing the object detection model to recognize multiple types of obstacles and incorporating further safety mechanisms, such as ultra wideband (UWB) based localization corrections would further improve the robustness. Additionally, expanding the system to support multi-UAV and swarm coordination tasks could enable sophisticated applications in real environments and more complex missions.

REFERENCES

[1] "Poyraz drone," HAVELSAN, [Online]. Available: https://www.havelsan.com/tr/urunler/poyraz-insansiz-hava-araci. [Accessed: Apr. 17, 2025].

[2] Y.-b. Ai, T. Rui, X.-q. Yang, J.-l. He, L. Fu, J.-b. Li, and M. Lu, "Visual SLAM in dynamic environments based on object detection," *Defence Technology*, vol. 17, no. 5, pp. 1712–1721, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214914720304402

[3] C. Sun, W. Zhan, J. She, and Y. Zhang, "Object detection from the video taken by drone via convolutional neural networks," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, Oct. 2020. [Online]. Available: https://doi.org/10.1155/2020/4013647

[4] M.-F. R. Lee and S. H. Yusuf, "Mobile robot navigation using deep reinforcement learning," *Processes*, vol. 10, no. 12, p. 2748, Dec. 2022. [Online]. Available: https://doi.org/10.3390/pr10122748

[5] G. Jocher and J. Qiu, "Ultralytics YOLO11," version 11.0.0, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[6] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," *arXiv preprint arXiv:1405.0312*, 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[7] IsaacLab Project Developers, "IsaacLab: Unified framework for robot learning built on NVIDIA Isaac Sim," 2025. [Online]. Available: https://github.com/isaac-sim/IsaacLab. [Accessed: Apr. 6, 2025].

[8] "Depth camera D435if," *Intel® RealSense™ Depth and Tracking Cameras*, [Online]. Available: https://www.intelrealsense.com/depth-camera-d435if/. [Accessed: Apr. 17, 2025].

[9] PX4 Development Team, "PX4 User Guide (v1.11)," *Dronecode Project*, 2021. [Online]. Available: https://docs.px4.io/v1.11/en/. [Accessed: Apr. 6, 2025].

[10] "F450 Quadcopter Image," [Online]. Available: https://www.google.com/search?q=f450+quadcopter. [Accessed: Apr. 6, 2025].

BEHIND THE SCENES