Autonomous Drone Grasping

Hairui Yin
Hsing-Hao Wang
Javad Baghirov

Our Objective

- Our goal is to develop a drone-based delivery system that is capable of autonomously picking up and transporting packages
- Specifically, it contains three skills:
  - Takeoff and Landing
  - Navigation and Obstacle Avoiding
  - Detecting and Catching Packages
- The whole process should be like: The drone takes off, identifies the package’s location, flies toward and hovers above it, the drone will descend until the magnet makes contact, the drone will then take off and deliver it to the destination or returns to the starting position.

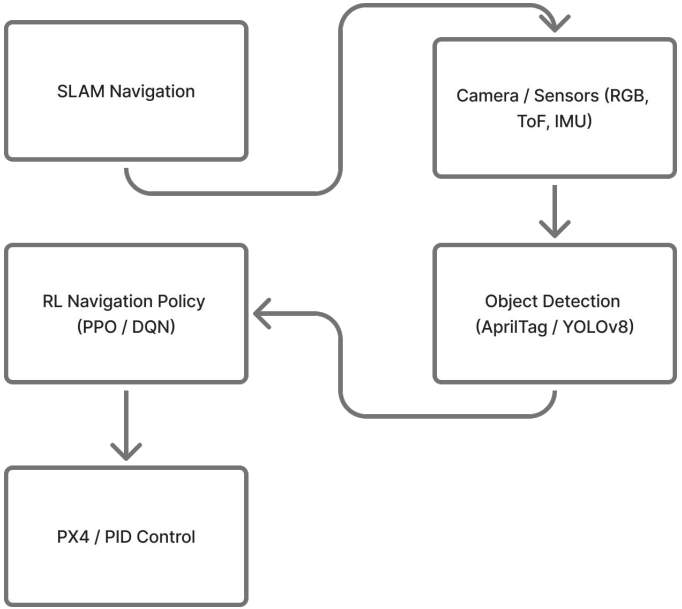
Hardware and Software

- Starling 2 Max GPS-denied Development Drone
  - Qualcomm QRB5165: CPU, GPU, NPU
  - 8GB LPDDR5
  - Dual image sensors: One faces forward, One faces down
  - Able to carry 500g weight
  - 380mm*450mm*120mm
  - VOXL2 SDK:
    - Voxl-px4 package to run service on Linux
    - Supports **Gazebo** Simulator
    - Programming: Python && C++

Control System Architecture

- High-Level Control:
  - RL algorithm: PPO (Proximal Policy Optimization)
  - For Stable Flying, Path Planning, Navigation, Grab Control

- Low-Level Control:
  - PX4 Controller
  - Receives high-level instructions from the PPO strategy, operates at a very high frequency, calculates instantaneous motor instructions, and ensures that the drone stably executes high-level instructions



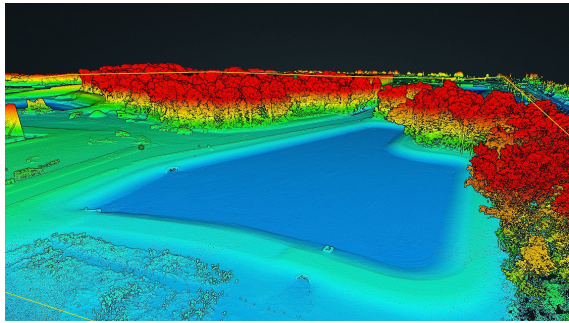
SIM2Real
ML Design

Takeoff and Landing

- Take off at starting point
  - Initialize RL-based take off policy
  - Monitoring drone altitude, stability, and drift
  - No camera input required
- Once airborne, start navigating the environment
  - Switch policy to navigate the environment
  - Turn on camera for visual input
- Return to starting point and land the drone
  - Switch into landing sub-policy
  - Monitor drone altitude and throttle input, maintain descent speed
  - Once in contact with ground, turn off motors

Navigation and Obstacle Avoidance

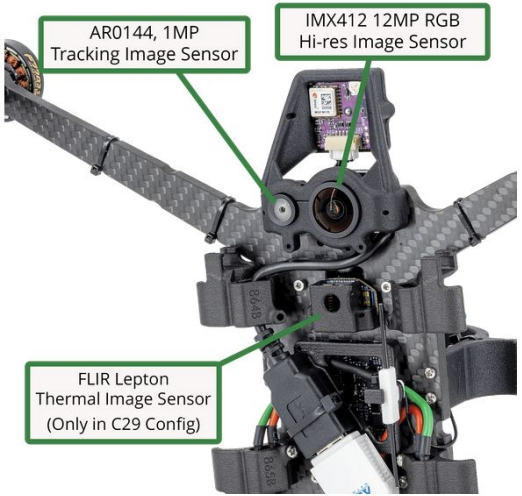
- Use a Front Monocular Camera, No LIDAR
  - Two parts: Visual and Policy
- Visual Input
  - The Target of this part is to convert image signal to depth map or cloud map
  - Depth Map: CNN, Transformer
  - Cloud Map: VGGT-Visual Geometry Grounded Transformer
- Policy
  - The Target of this part is to convert our processed visual signal to policy
  - Actor-Critic: PPO
- Extra Thoughts:
  - Build Memory of the map with cloud map, and plan the route from high level



Cloud Point

Object Detection

- Front and Bottom Cameras - “IMX412 12MP RGB”
- Once the drone takes off the item will be visible on the camera
  - Option A: Use YOLO with real time inference to detect the item
    - Training would be images of the item from different positions with different lighting
  - Option B: Use HSV color segmentation to detect the package. The package will always be 1 color and the system will detect the package by identifying that color in the camera feed



Object Catching

- A magnet attached with a rope will be mounted to the bottom of the drone.
- Using the bottom IMX412 12MP RGB camera, the drone will detect the target object and align itself directly above it.
    - Check the distance between the center of the bounding box to the center of the image
- Once centered, the drone will descend slowly until the magnet makes contact and attaches to the metallic object.

Reference

- https://docs.modalai.com/starling-2-max-datasheet/
- https://docs.modalai.com/voxl2-PX4-hitl/
- https://docs.modalai.com/mavsdk/
- https://arxiv.org/abs/2411.03303
- https://arxiv.org/abs/2503.14352
- https://arxiv.org/abs/2503.11651
- https://ardupilot.org/copter/docs/common-modalai-voxl2.html