CyDrone

Team sdmay20_47

Kenneth Lange, Alain Njipwo, Daniil Olshanskyi,
Luke Bell, Max Medberry

Client and advisor: Dr. Ali Jannesari

Problem and motivation

- Everything is being automated
- Calculating powers are available and small
- Neural networks do great and picture analysis
- Stereo cameras can give precise depth images
- Drones are a widespread and powerful tool but require an operator

Project idea

- Put computation powers ON the drone
- Make the computer drive the drone
- Make it autonomous
- Allow complex object analysis algorithms (like volumetric analysis)

Figure 1: Volumetric Analysis

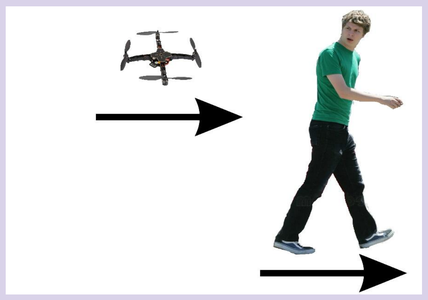


Figure 2: Autonomously Follow Target

Functional requirements

- ❖ Incorporate information about old iterations of the project received from the client
- ❖ Set up a simulation environment to test the software.
- ❖ Implement object detection and compute the volumetric analysis
- ❖ The hardware will be Master-Slave oriented so that new hardware can be added and removed as desired in the future
- ❖ Assemble the drone itself
- ❖ The onboard computer should be able to control the drone movements based on pictures camera takes

Non-functional requirements

- ❖ All software should store logs of the past important information and crash reports
- ❖ Software will have an architecture that is easy to understand and navigate
- ❖ Code will be well documented, both via comments and a wiki explaining to future users what is being accessed by what and how
- ❖ Code will be modular, so it is easy to fix, extend, and/or replace
- ❖ Open source software will be used to make the process cheaper and more maintainable using the help of an open source community

Operation environment

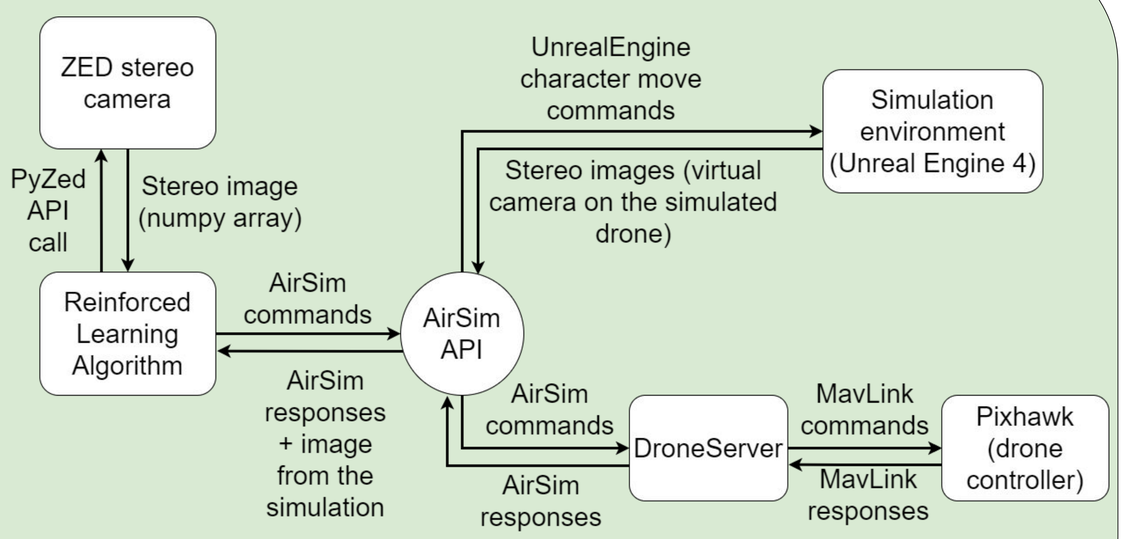
- ❖ Drone will be operated outdoors and indoors
- ❖ User may not be an experience drone pilot
- ❖ Wind may be present

Intended users

- ❖ Industry and farms (volumetric analysis)
- ❖ Bloggers (drone-companion)
- ❖ Police (drone-seeker)
- ❖ Extendable by other developers

Design sketch

- ➔ RLA should operate on either real camera images or simulated ones (transformed to the same format)
- ➔ Commands produced by RLA are of the same format (AirSim commands)
- ➔ RC controller should be able to intervene as a fail safe measure



Hardware Standards:

- PX4 flight stack and ArduPilot
- CAN
- MavLink protocol

Software Practices/Standards:

- GIT
- AirSim API
- Client-server
- ROS

Software Testing:

- Testing simulation flight
- Verifying dodging obstacles
- Ensuring distance from target

Hardware Testing:

- RC control test Flights
- Drone RC control test (no blades)
- Sensor Calibration
- Zed camera diagnostics test

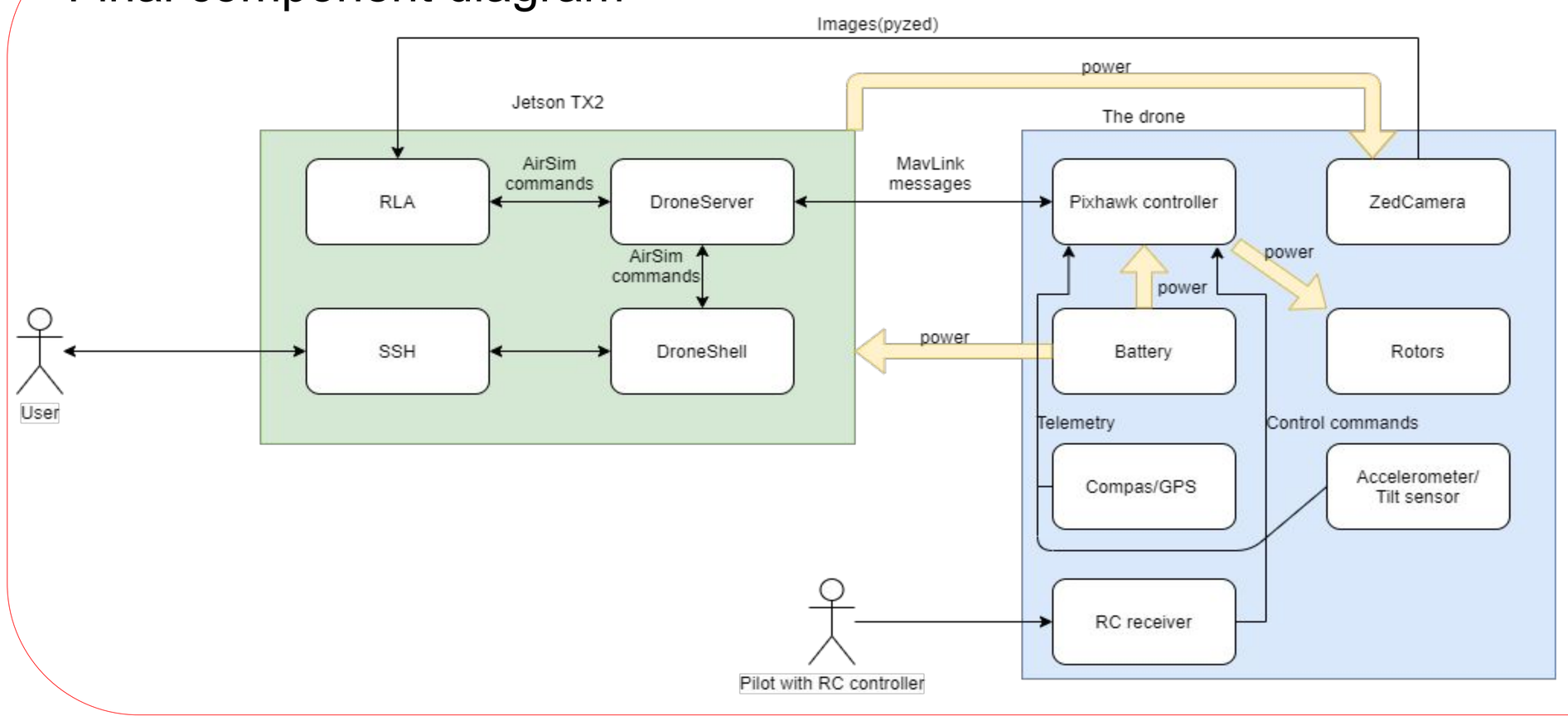
Testing Strategies:

- Functional Test System
- System Testing
- User Acceptance Test
- Client Certification Environment

Testing Environment:

- Ubuntu Titan Computer (Simulation Environment)
- Howe Hall (Test Flights)

Final component diagram



Technical details:

- Can control drone from RLA, from SSH, or from RC
- LiPo battery powers Pixhawk and Jetson. Pixhawk powers rotors. Jetson powers ZED
- RLA gets images from ZED camera, processes them and decides which AirSim calls to make
- Pixhawk firmware ensures stable flight
- Pixhawk responds to both RC controller and whoever “listens” on Jetson