

Development of Software-in-the-Loop Simulation using Airsim and Px4 for the Unmanned Aerial Vehicles

Sivaramakrishnan Rajendar¹, Moulis K², Mukilan S R³, Nagul Pranaw⁴

Abstract

The popularity of autonomous vehicles and their demand are on the rise. But, the development and testing algorithms for autonomous vehicles are costlier and time consuming one. In recent decade, the Unmanned Aerial Vehicles (UAVs) are widely used in many fields such as civil, military, and delivery purposes, etc. Therefore, it is essential to test the features of UAV before its deployment. The complexity of the software controlling and testing has also been grown in recent years. The simulator built on Unreal Engine provides physical and visual realistic simulation environment for the users. It contains a physics engine that operates at high frequency Software-In-The-Loop simulation. The simulation supports various popular protocols. Moreover, it is relatively easier to implement various components in other projects with the help of modular design. In the proposed model, the non-linear 6DOF model 3D visualization and sensor model are implemented using Airsim for the quad-rotor UAV. In addition, virtual Pixhawk hardware is used for the design and implementation of the flight control algorithm. Verifying through a simulated environment increases the safety concerns and thus reduce cost. The demonstration involves a quadcopter as an autonomous vehicle.

Keywords: Aerial Vehicles, Drones, Unmanned Aerial Vehicles, Autonomous Vehicles, Airsim, Unreal Engine.

¹Assistant Professor (Sr.G), Department of CSE, KPR Institute of Engineering and Technology, Coimbatore, sivaramakrishnan2010@gmail.com

^{2,3,4}UG Scholar, Department of CSE, KPR Institute of Engineering and Technology, Coimbatore.

²17cs091@kpriet.ac.in, ³mukilcoolstar@gmail.com, ⁴nagule8@gmail.com

Received Accepted

Introduction

Unmanned Aerial Vehicle (UAV) are widely being implemented because of their ability to perform tedious tasks such as Tracking, Delivering, Disaster monitoring, Surveillance, etc. The reinforcement learning and learning-by-demonstration have become essential in Unmanned Aerial Vehicle (UAV). However, the sample complexity is too high. The UAV doesn't only mean the drones, but also covers copter or rover and cars. UAV system comprises of a number of sub-systems, the Qground control station and the communication links [7].

Airsim is an open-source platform used for the development of autonomous vehicles and helps to bridge the simulation and the reality. Software-in-the-loop is a simulation tool that enables one to develop a testbed to avoid critical situations and help saving the costlier equipment from a damage. Qground control is used to develop the mission plan and the path depending on the environment and requirements for the UAV [1]. Further, the UAV has to attain the mission based on the determined obstacles.

The UAVs will be placed at various positions and altitudes during the mission to reach the

target zone. Moreover, the Qground control should oversee the payload parameters while the mission is on. It offers a GUI through which a user can monitor the status of the packet transfer over the network. A numerous sensors and camera are mounted on the UAV to serve the need.

Literature Survey

Home Delivery using Android

In 2014, M. Muhammad et al., [2] proposed an autonomous copter for supplying products to a home. It uses an android-device as a processing unit and is being connected to a flight controller. The systems utilizes Google map walking mode direction to identify the path. However, the path is a shortest one, as it depends on the ground path. Hence, it usually becomes longer than the actual straight path. It doesn't consider more than one destination. The system requires high processing power and installing the libraries is not possible only on mobile devices.

Quadcopter for Drug Shipments

Alberto Lisanti et. al. [3] introduced a quadcopter for delivering drugs in 2015. The application can be used by the clients and the pharmacies to request for medicines. Moreover, the application allows only one destination for the quadcopter, which implies that it accepts exactly one request at a time. In addition, there is exactly one path to return to the parent location. To make the travelling an optimized one, no load calculations are performed. The system also allows multiple destinations at the same time, an optimized driving mode. The guided mode enables the fully autonomous mode.

Quadcopter using GPS Module

Roberts, C et. al. [4] developed a quadcopter using GPS module autonomously. The Navio flight controller is user connected with RPI. There is no UI and remote access to the user to send requests. It doesn't involve energy estimation and effective path planning. The Qground control station is a non-military drone ground control station and can be connected to joystick and gamepad only. A naïve user can't easily modify the system. Moreover, it is impossible to increase the number of channels in the control systems. The software shall not display other information such as weather, drone temperature, sensor data. These information are essential for a safe drone flight operation. Proving intelligence to the software and the telemetry module is difficult, since the software is directly connected with the telemetry module and is continued. Thus, it is possible for the drones being hacked by any malicious remote operator.

Proposed System

The paper purposes to build a middleware to test the quad-rotor UAV operations and to secure SITL communications. In the proposed model, the non-linear 6DOF model, 3D visualization and the sensor model are implemented using Airsim. In addition, virtual Pixhawk hardware is used for the design and implementation of the flight control algorithm. Control Application Software (CAS) is a new middleware application which follows multithread structure to guarantee the connection and data transfer among the Airsim and Pixhawk. The GUI in the CAS enables a user to oversee the stage of packet transfer. Commanding flight control algorithm and parameter tuning for the rotor UAV. The effectiveness of CAS is proved using the numerical implementations.

The hardware part in the proposed system is not a readymade one similar to a input control system such as a joystick or gamepad. On the other hand, it is a customized input control system based virtual controller that can be reformed easily by the users. The bridge between the software and the input devises is this virtual controller. Airsim is used to write the main control

algorithm since a regular software relies on computers and OS. The entire station will collapse if the computer breaks down to function due to a power cut or damaged by the malicious third party operator. The virtual controller further allows more sensors to be added for monitoring purpose. Thus, the software is able to visualize the state of the drone including the voltage, temperature, etc. in addition to the telemetry data.

System Architecture

In this system, the Qground control gives the final output as an Ulog file. So that we can able to know what are the pros and cons that affects the drone during the simulation.

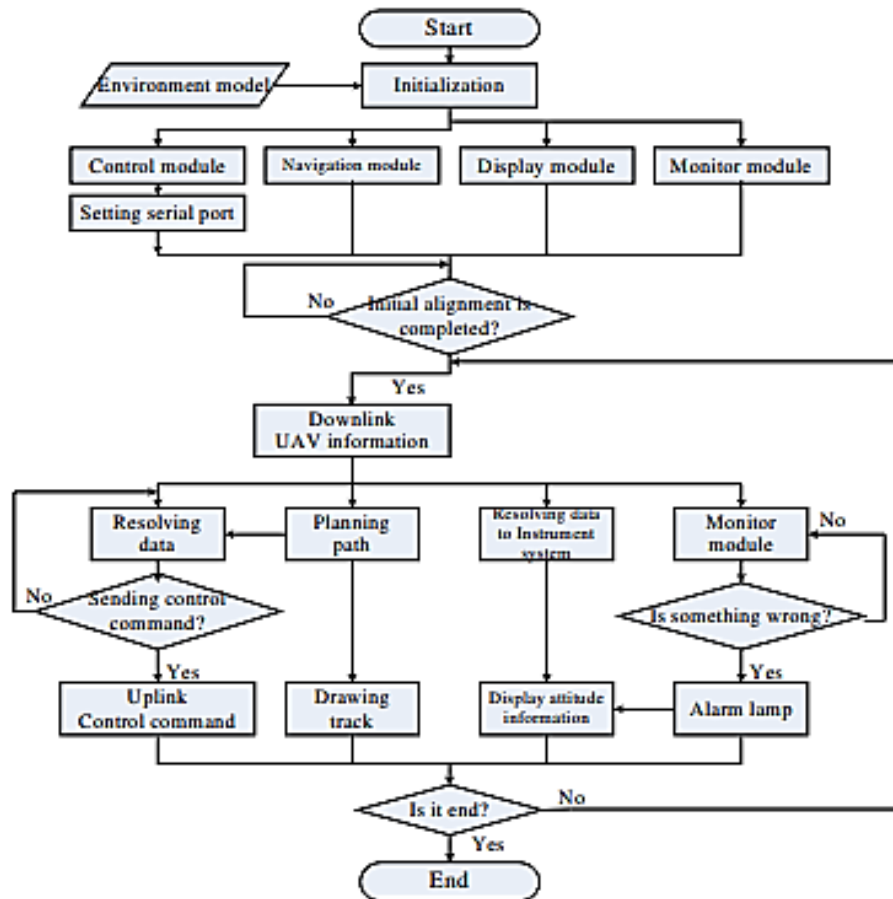


Fig. 1. System Architecture

In the above system represented in Figure 1, the simulation environment is get initialized after that, PX4 virtual controller is get compiled in the command prompt. The Qground control station helps to assign path to the Quadcopter. Unreal engine helps to visualize the simulation. Data is transferred to Qground control through PX4.

Results and Discussion

In this system by using the Airsim platform and Px4 virtual controller, we can control and monitor the performance and the response of the drone [5]. The results are obtained on a log file. The log file includes sensor information, path, frequency and bandwidth of the drone which can be analyzed further for future enhancement. The following command is used to build PX4 firmware and the Figure 2 represents the same.

make px4_sitl_default none_iris

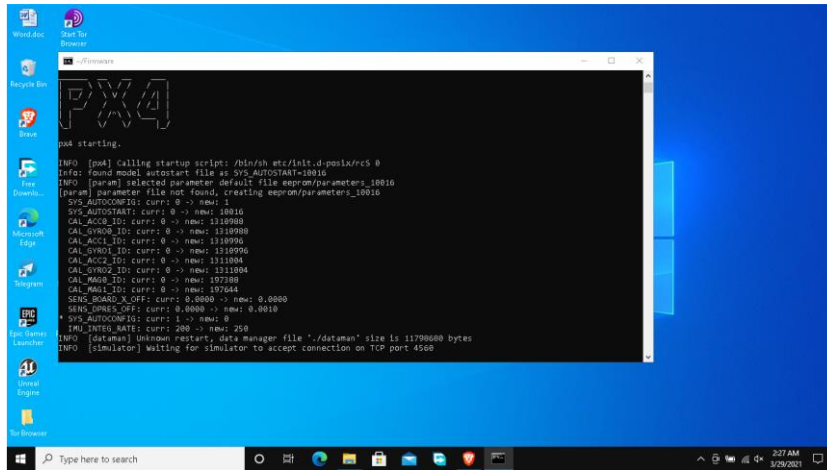


Fig. 2. Building PX4

After the build is complete it will show a message like “waiting for simulator to accept connection on tcp port” as shown in Figure 3 [6].

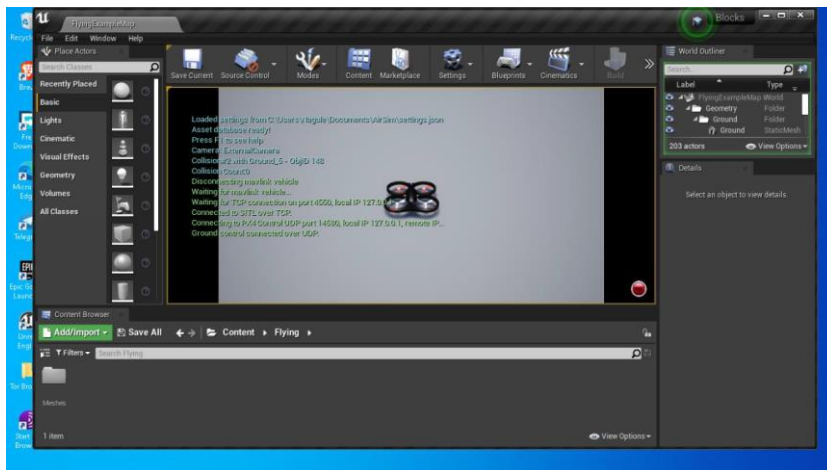


Fig. 3. Connection to TCP port

The Figure 4 shows the ground control which helps to assign path for the drone [8] [9] [10].

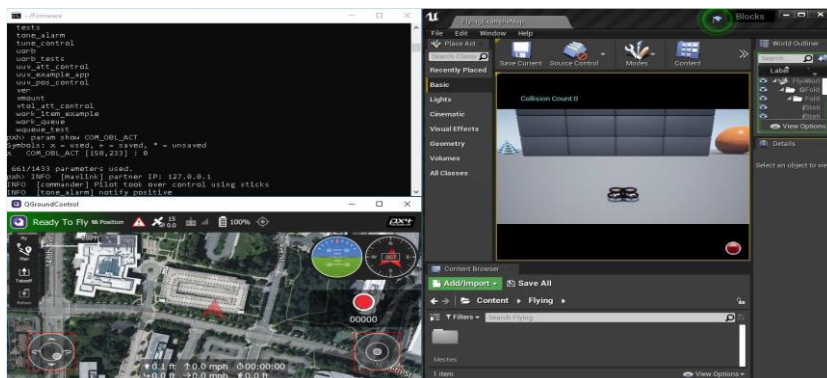


Fig. 4. Ground control

Conclusion

The implementation of an autonomous quadcopter using SITL was carried out in this

paper. Qground controller is used to control the quadcopter by instructing Airsim drone controller with the help of Dronekit API. The Dronekit API sends MAVLink messages to PX4. This enables the PX4 to provide intelligence for image processing and trajectory planning. A flight controller cannot carry out this standalone. The experimental results are promising while landing with 3 to 5 meter difference to a parent location or any other. It is due to the errors in the GPS signal. Such errors shall be reduced by incorporating precision landing techniques to the quadcopter. The paper demonstrated the behavior of a quadcopter such as takeoff, landing, and navigating to the desired location autonomously based on the user input through the interface.

References

- Syed Omar Faruk Towaha, Clerk Maxwell “Building Smart Drones with ESP8266 and Arduino”, A Treatise on Electricity and Magnetism, BIRMINGHAM – MUMBAI, 2018.
- M Muhammad, D Swarnaker, and M Arifuzzaman, “Autonomous Quadcopter for Product Home Delivery”, International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) 2014.
- Alberto Lisanti and Giorgio Venezia, “New Frontiers of Delivery Services Using Drones: a Prototype System Exploiting a Quadcopter for Autonomous Drug Shipments”, IEEE 39th Annual International Computers, Software & Applications Conference 2015.
- Cameron Roberts, “GPS Guided Autonomous Drone”, University of Evansville, College of Engineering and Computer Science, April 25, 2016.
- ArduPilot software documentation [Online]. Available on: <http://ardupilot.org/about>. [Accessed 2 April 2019].
- DroneKit by 3DR Robotic [Online]. Available: <http://dronekit.io> [Accessed 5 April 2019].
- Eric N. Johnson and Daniel P. Schrage, “The Georgia tech unmanned aerial research vehicle: GTMax,” School of Aerospace Engineering, Georgia Institute of Technology, Atlanta 2003.
- DroneKit-Python documentation [Online]. Available: <http://python.dronekit.io/1.5.0/about/overview.html> [Accessed 7 April 2019].
- MAVLink common messages set and documentation [Online]. Available: <https://mavlink.io/en/messages/common.html> [Accessed 3 April 2019].
- MAVProxy documentation [Online]. Available: <http://ardupilot.github.io/MAVProxy/html/index.html> [Accessed 15 March 2019].