Swarm UAVs for Area Mapping in GPS-Denied Locations

Daniel Golan[1], Patrick Kennedy[1], Bryan Gonzalez[1], Ryan Taylor[1], Joseph Perry[1],Ethan Thomas[1], Ryan Ebrahimi[1], Kyle Fox[1], Sergey Drakunov[1]

[1]Embry-Riddle Aeronautical University, Engineering Physics Propulsion Lab, 1 Aerospace Blvd, Daytona Beach, FL, USA 32114

ABSTRACT

Swarm of Unmanned Aerial Vehicles using Emergence (SUAVE) efficiently creates a map in GPS-denied locations using Simultaneous Localization and Mapping (SLAM). Each drone is outfitted with a Lidar and an Intel Realsense Camera, which will be used to develop a high-fidelity infrared 3D map of an area without the need for GPS. Upon completion of each flight, the point clouds made through photogrammetry on each drone will be recorded and fused, creating a map with higher precision and accuracy. The use of the swarm is to generate this map from multiple points of view so that shadow effects can be negated and a more populated, dense map can be produced in respect to one drone's map. To address localization with no GPS, the onboard IMU suite will be used to track relative position, while the onboard camera will track the local position. These two forms of localization, when coupled, allow for autonomous flights in lieu of GPS localization. This manuscript demonstrates the differences between one, three, and five fused maps during autonomous test flights on an Unmanned Aerial System (UAS) with a lack of GPS.

Keywords: Swarm Robotics, SLAM, Photogrammetry, Point Cloud, UAS, GPS-Denied, Sensor Fusion, Lidar

1. INTRODUCTION

Swarm robotics is the study and control of multiple independent entities with the goal of achieving a collective behavior toward completing a common task. This behavior is achieved through the order of execution of its tasks as well as the minimizing of resources per robot to complete the overarching goal while keeping each robot distinctly independent. These characteristics can be seen in various facets of nature where groups of organisms can solve issues too large for any single element to manage. The original theory behind swarm robotics began with G. Beni, who theorized a robotic system based on a cellular space where autonomous robots had limited communication and no central point of control but achieved complex goals[1]. This theory is based on organic cells, which describe the number of nodes in a swarm as greater than 100. The study was based on simulations of smaller objects in greater numbers. The driving factors behind what can be considered a robotic swarm are limiting the communication of each item to remain local, working together to achieve a common task, limiting the amount of human control, and allowing the swarm to act in a semi-autonomous manner[2]. Throughout the past two decades, there has been a surge in swarm intelligence research as the applications of this emerging avenue of robotics are endless. One of the desirable features of swarm robotics is the increased flexibility of the system to adjust to its environment.

The localization method used in this project is known as Simultaneous Localization and Mapping (SLAM). SLAM programs create three-dimensional point clouds through the use of various sensors and then localize themselves within their environment based on the spatial map they generate throughout the flight. This can be used to find their local position without the use of GPS and is highly favorable for indoor applications. The use of SLAM and photogrammetry coupled with onboard inertial measurement units (IMU) for relative position gives accurate localization in lieu of GPS.

2. METHODOLOGY

The autonomous control of the drone and the mapping were tested and executed on a set of two bespoke quadcopter UASs, as seen in Figure 1. Each is equipped with a Lightware lidar SF45b and an Intel RealSense d455 camera for use in creating the 3D map of the target environment. Each drone is fastened with a LattePanda 3 delta companion computer for SLAM computing and autonomous navigation. A Pixhawk Cube Orange is employed as the flight controller, which

Unmanned Systems Technology XXVI, edited by Hoa G. Nguyen,
Paul L. Muench, Robert Diltz, Proc. of SPIE Vol. 13055, 130550O

executes commands sent by the companion computer based on sensor feedback and initial starting functions. The communication between the Pixhawk and the companion computer uses MAVlink.

Figure 1: UAS with Camera and Companion Computer

The flight computer locates the original position of the drone through an extended Kalman filter that defaults to the IMU and GPS data. This process eliminates propagation error due to IMU drift. However, this fails in GPS-denied locations; therefore, the flight computer requires a secondary source of global position and pose to eliminate the IMU drift[3]. In this case, RTAB-Map publishes an ROS (Robot Operating System) topic with the drone's real-time odometry and updates the flight computer's Visual Inertial Odometry (VIO) using an MAVLink message. MAVLink (Micro Air Vehicle Link), is a communication protocol designed for sending messages between the flight computer and a ground station or companion computer. On the companion computer, the RTAB-Map odometry is converted into VIO and odometry MAVLink messages and sent to the flight computer. Because the flight computer is provided with odometry from both the onboard IMU and the SLAM algorithm, the extended Kalman filter can converge onto a local position for the drone, negating the need for GPS.

To perform the SLAM algorithm, a LattePanda 3 Delta single-board computer was used as the onboard computer to meet the computing requirements needed to run the Realsense ROS node and RTAB-Map ROS node. RTAB-Map is a visual approach to SLAM, meaning that it uses the visual information collected by the RGB-D Realsense camera to collect data for mapping an environment and keeping track of the drone’s location within that environment. The algorithm primarily relies on loop closure detection to recognize points of interest that correspond to previously visited areas of the map, which is used in conjunction with the position estimate calculated by dead reckoning data collected by the onboard IMU to continuously update the position estimate. The recognized points, or nodes, used for loop closure are determined using OpenCV for computer vision, and this method of using nodes to represent the spatial areas captured in the frames along with the edges to represent the movement or transformation between these nodes is known as a graph-based approach to map maintenance. This method allows for efficient handling of large-scale mapping of long-term data as it allows for the selective merging of new nodes with existing data, maintaining the consistency and accuracy of the map over time.

The odometry provided by RTAB-Map is within the camera link frame, which is a globally fixed frame aligned with the front-left-up frame of the drone at the moment RTAB-Map is launched. The camera link frame is an arbitrary frame relative to the Pixhawk; sending odometry in this frame would lead to failure as the VIO will update the Pixhawk's position in the wrong direction. To compensate for this, the heading of the drone at the time the camera link frame is initiated will be collected. A one-dimensional rotation about the Z axis will be applied using this heading to convert the camera link frame into the North East Down (NED) frame. Updating odometry in the NED frame is accepted by the Pixhawk, allowing the readings from the VIO and onboard IMU to converge after the rotation is applied.

In addition to updating the odometry of the flight computer, MAVLink is used to control the motion of the drone. Given a valid position estimate from VIO, PX4, the firmware for the Pixhawk, will allow switching modes to offboard mode. Offboard mode is the flight mode used to send autonomous commands from the companion computer to update the position in a NED frame which the flight computer will execute.

The complete system architecture is outlined in Figure 2.

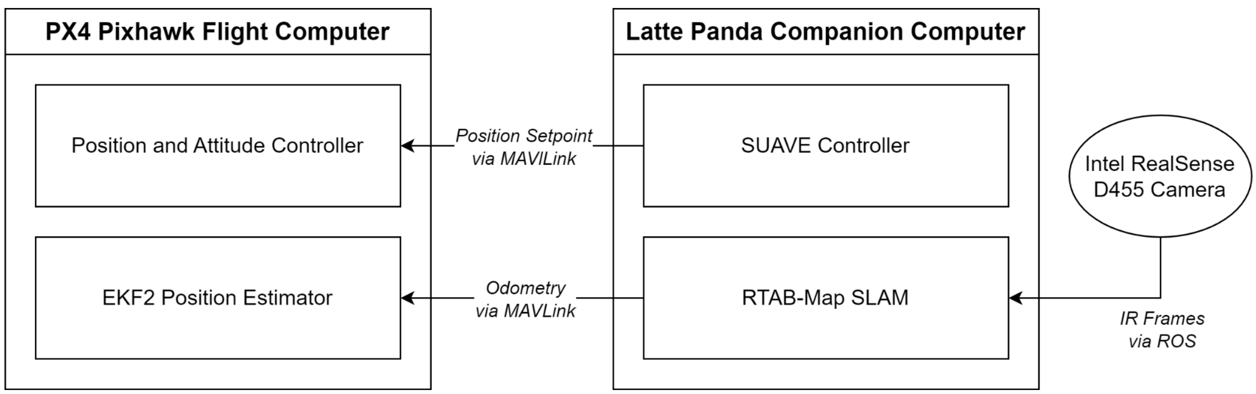

Figure 2: System architecture for autonomous drones utilizing VIO for position estimation.

The SUAVE Controller, referenced in Figure 2, is a system onboard the Latte Panda that executes mission goals and evaluates targets. For testing purposes, the controller was programmed to achieve two goals. The first focuses on pitch control, and having the drone fly 1.5 meters on the -Z axis, translate 1.5 meters in the +X axis of the drone, then land. The second goal is to focus on roll control; the drone should fly 1.5 meters in the -Z axis, translate 1.5 meters in the -Y axis, and then land. The tests are intended to evaluate the pitch and roll performance of the drone as it receives commands and setpoints from an offboard controller while using VIO for position estimation.

Once the flight is completed and the data has been collected, it will be fused. The post-flight fusion uses the point clouds created by RTAB-Map and saves it as a polygon file (.ply). These .ply files are modified and merged using CloudCompare, an open-source visualization and cloud editing tool, in order to create a much more accurate map. Fusing the UASs points of view allows for higher precision and accuracy for the final map while taking less time than if a single drone were to attempt this. The maps are fused by aligning them roughly then applying a transformation matrix to align them. They are then fused, and the process is repeated with multiple point clouds until the desired point density and accuracy is achieved.

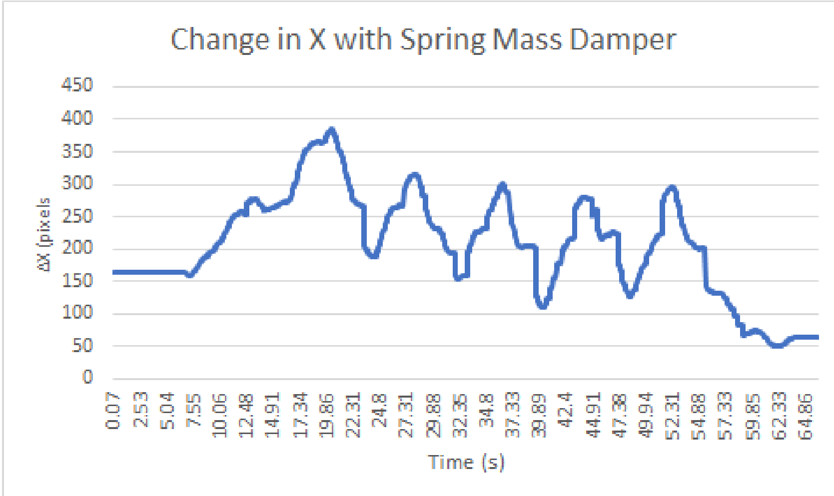

Figure 3: Tello Spring-Mass Damper PID Controller

Figure 3 shows the data from the original swarm PID made using DJI Tello Drones. The Tello drones were the original basis for the swarm; however, they were not as applicable past the proof of concept due to their lack of controllability. This is evident due to the lack of convergence in Figure 3 because of the inability to change position at a distance below 20 cm per movement. A swarm was not attempted on the larger UAS shown in Figure 1; however, the Spring-Mass Damper that was made for the Tellos will be applied to them for maintaining their distance between each entity and for object avoidance.

3. RESULTS AND ANALYSIS

Regarding dynamics and controls, both the pitch and roll test behaved as expected. The PX4 flight log data was stored for each flight and is displayed in Appendix 2. For the pitch-only test case, the drone performed a change in pitch to move a specified distance autonomously in offboard mode while using VIO for local position estimation. When the initial takeoff instruction was sent to the Pixhawk, the setpoint for the Z position was dictated, and the drone increased thrust until it converged on that specified position. To confirm that the drone reached the set altitude, a delay was added before any X or Y translation commands were sent. To reach the desired X position, the translation commands were sent in steps to control the rate at which the drone moved to the point. If the final point was set with one translation command, the drone would accelerate too quickly which would lead to excessive overshoot and loss of position tracking from RTAB-Map and VIO. The desired point was 1.5 meters away in the +X direction and was broken up into three 0.5-meter translation commands with delays in between. These delays slowed down the horizontal velocity to help avoid any overshooting of the desired point or tracking issues. Figure 16 shows the X setpoint with the estimated local position on the X-axis to see the commands executed in a position setpoint. Figure 14 shows the pitch angle setpoint and the estimated pitch angle of the drone to show how the drone reached the setpoints. For each translation, the drone experiences a large initial angle change to gain momentum, then goes back to zero before slightly overshooting zero to tilt back the other way and stop its momentum. The estimated pitch angle very closely follows the setpoint as the pitch axis was well-tuned and experienced minimal vibrations. In the roll translation flight, the similarity between setpoints and estimated angles are more varied. Figure 10 shows how the roll angle setpoint varies much more in comparison with the pitch angle setpoint in Figure 14. This is likely because the roll axis of rotation was not as stable as pitch and experienced more vibrations. Before the successful flights, the roll axis was so unstable the drone was uncontrollable. These vibrations were fixed by turning down PID response rates in the roll axis but could be further tuned to optimize accuracy in the roll angle. The difference in stability between roll and pitch angles can also be seen in the angular rates for roll and pitch (figures 11 and 15). Also, the scale of the angle in the angle setpoint and estimated angle graphs (figures 10 and 14) should be noted to compare the variation in setpoints.

After the test flight, the odometry predicted by RTAB-Map was captured. For the pitch and roll translation flights, the position of the camera from RTAB-Map is shown in Figures 4 and 5, respectively.

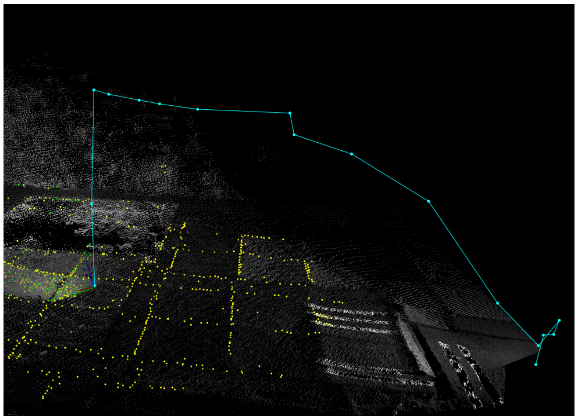
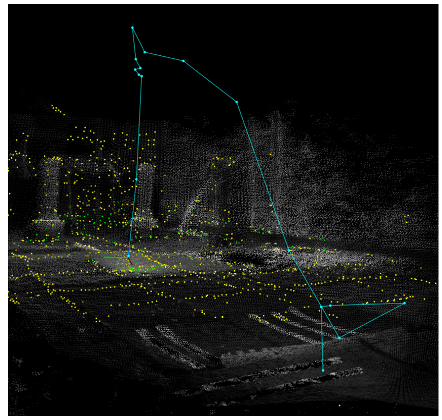

Figure 4: Position (cyan) predicted from RTAB-Map for pitch translation flight.

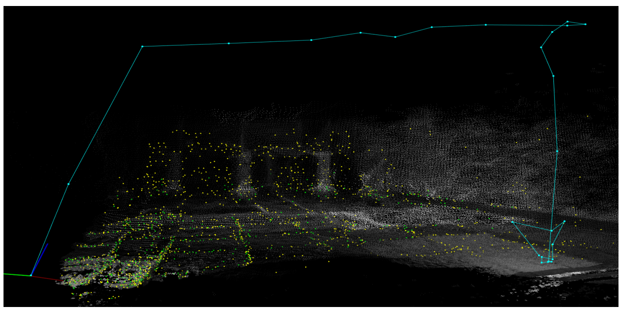
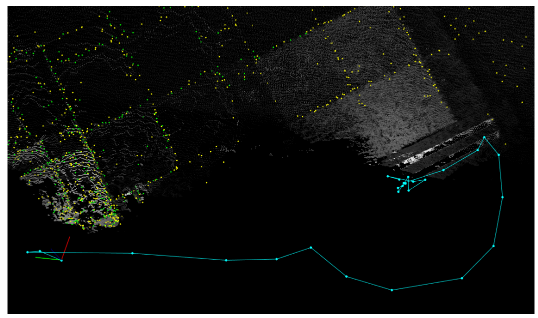

Figure 5: Position (cyan) predicted from RTAB-Map for roll translation flight. Top-down view on the right.

By comparing Figures 4 and 5 to the flight log position data, it can be deduced that the SLAM algorithm is suitable for predicting the actual odometry of the flight. In addition, the algorithm had a high enough fidelity that the drone could use its data to correct for errors in areas of instability such as takeoff. Figure 4 shows the correct flight where the drone took off, translated in +X, and then landed. Figure 5 shows the correct flight where the drone took off, translated in -Y, and then landed. It should be noted that the top-down view in Figure 5 shows an area of instability where the drone drifted before the first translation instruction was executed. During this translation, the drone corrected its position and converged on the setpoint.

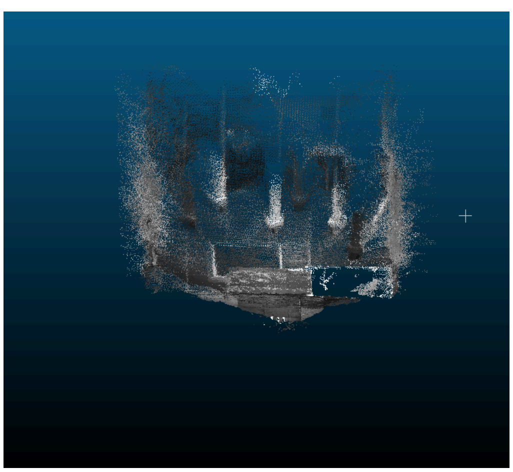

Figure 6: 1 Point Cloud

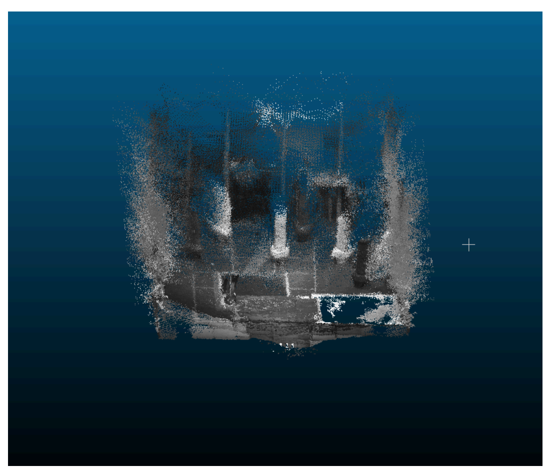

Figure 7: 3 Fused Point Clouds

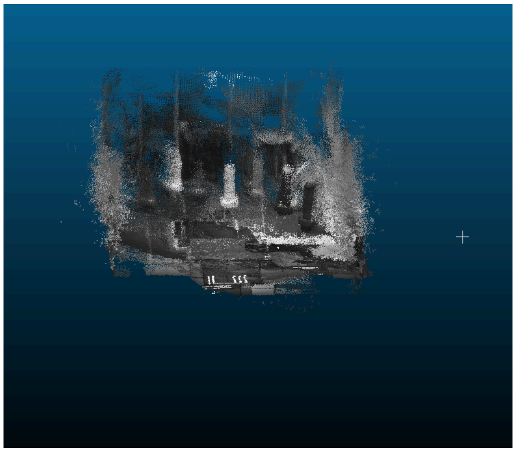

Figure 8: 5 Fused Point Clouds

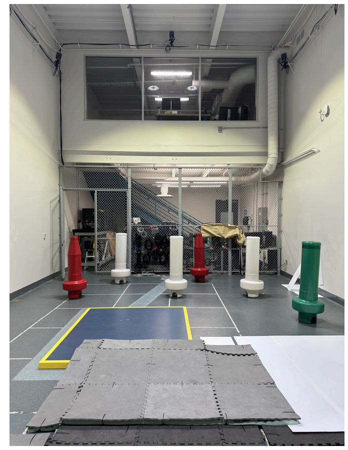

Figure 9: Picture of testing facility

Each time the UAS is flown, a point cloud is collected from RTAB-Map using photogrammetry. After the flight, each map is stored and uploaded to a shared drive to be modified and fused. Figure 9 shows the testing facility that was being mapped for Figures 6, 7, and 8. The buoys were placed in an arbitrary pattern with varying colors to test the effectiveness of different types of point-collection methods. The first two methods attempted were stereography and color mapping, however, the sampling rate was too low, and RTAB-Map would fail with the drone flights. The maps above are in infrared. Infrared (IR) proved to have the highest sampling rate and was able to handle quick movements without losing the VIO stream, making it ideal for this application. One issue with the IR mode was the lack of color with the produced point cloud. The camera was able to register the white buoys well and the red buoys as a bit darker, but the green buoy is significantly darker. The tests were completed indoors so the camera would not be overexposed when outdoors, and the walls gave a stark outline to the populated area on the floor. The mats on the floor not only acted as a safety measure but also added texture so RTAB-Map could be more accurate in its created environment[4].

Figure 6 shows the point cloud generated from an individual flight for the duration of seven seconds. The produced point cloud has a high fidelity, however, it is missing the top of the buoys, and there are large areas where no data was collected, specifically the ground and the back wall. Figure 7 uses three-point clouds from three different flights, following the same testing process as the first setup. In this map, there are still open areas, but the objects are more defined, and the contrast between objects can be seen clearly. It also extends higher on all walls, which is due to the variation of the flight path due to IMU drift and slight shifts in the position hold of the drone. Figure 8 demonstrates a points cloud with five different flights combined. Although the floors and objects are denser, the surplus of point information saturates the cloud and some items begin to merge together. The green buoy on the far right began to merge with the wall next to it caused by too much information.

4. CONCLUSION AND FUTURE WORKS

With the use of the Pixhawk Cube Orange, a drone equipped with an Intel RealSense d455, Lightware lidar SF45b, and a LattePanda 3 delta companion computer, a system was developed that enables the drone to autonomously operate in GPS-denied areas. This was achieved by running the RTAB-Map SLAM algorithm while in flight to identify the local odometry of the drone, this odometry was sent to the Pixhawk to update the position estimator, which enables offboard commands to set the local position. During testing, it was found that the drone converges accurately on its target setpoint with some expected drift.

The performance of the drones were analyzed using the post-flight log data in the PX4 flight log data viewer. These results revealed copious amounts of information about how the drone was responding to control input, vibrations, or other variations throughout the flight which led to the addressing and fixing of many errors. The drone was able to successfully fly autonomously without the use of GPS while creating a map. After multiple flights, the maps were fused and it was shown that there is a window of the proper amount of point clouds to create a dense but not oversaturated map.

To improve the results of this project, the stability of the drones can be tuned to a higher degree, the drone frame hardware can be upgraded to handle vibrations with less wear, the position controls can be made more robust, and the implementation of swarm drone controls can be tested. Firstly, the tuning of the drones can be improved. The stability of the drones is a limiting factor in the accuracy of the positioning and mapping capabilities. Currently, the PID tune for the UAS is based on default values and arbitrary adjustments based on visual errors. With the IMU data, the PID values can be automatically tuned in post to vastly improve accuracy. The vibrations and instabilities in flight may also be caused by poor-quality hardware for the frame. Several points of failure have been identified and observed to be directly related to mounting issues, varying angles, and poor impact resistance. Some parts have been replaced with 3D-printed components either to accommodate for mounting points or to add strength and stability. A higher-quality frame will also be tested to eliminate any other hardware errors. This will yield a more stable flight, more responsive controls, and less position drift. In terms of position controls, the current setup sets a position and allows the drone to head to the set position at the maximum speed allowed in the PX4 parameters. This limits the ability to control velocity in any direction to delays and stepwise inputs. With the ability to set velocity, the rate at which the drone approaches the desired position can be turned down, which will also decrease the maximum angle of the drone and prevent the SLAM algorithm from losing VIO tracking. Velocity control can also help smooth the path between points and limit any unnecessary corrections.

Currently, only one drone is being flown at an instant using the autonomous VIO setup. A swarm of drones will greatly improve the mapping capabilities as many drones will be able to continuously create their individual map of the terrain that can be compiled into a high-fidelity model of the area they explore. Since GPS-denied autonomous position control has been achieved, to accomplish a swarm flight, a controller needs to be developed to instruct each drone on where to position itself. Difficulties in completing this include setting up a reliable communication protocol between each of the drones as well as defining a global coordinate frame in which each drone can define itself in. Without the ability to unify the coordinate frame each drone is in, many points of failure are introduced when attempting to fly a swarm controller. Solutions to this include determining the distance between two drones at any instance in time or identifying a common point in the SLAM data as the origin of the NED frame. In the future, there is a plan to use dual quaternions for swarm path planning in regards to each of the individual drones as well as the movement of the entire swarm.

Appendix 1

Terminology and Nomenclature

MAVLink	Micro Air Vehicle Link, communication protocol between companion computer and flight computer
MAVSDK	Library to send MAVLink messages to flight computer
VIO	Visual Inertial Odometry
SLAM	Simultaneous Localization and Mapping
UAS	Unmanned Aerial System
ROS	Robot Operating System
RGB-D	Red Green Blue Depth
NED	North East Down Frame
PX4	Pixhawk Firmware
RTAB-Map	Real Time Appearance Based Mapping
PID	Proportional Integral Derivative
IR	Infrared

Appendix 2

Flight Logs

Figure 10: Roll angle from roll translational Flight

Figure 11: Roll Angular Rate from roll translational Flight

Figure 12: Local Position Y from roll translational Flight

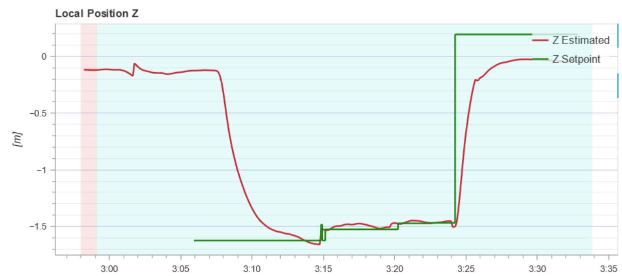

Figure 13: Local Position Z from roll translational Flight

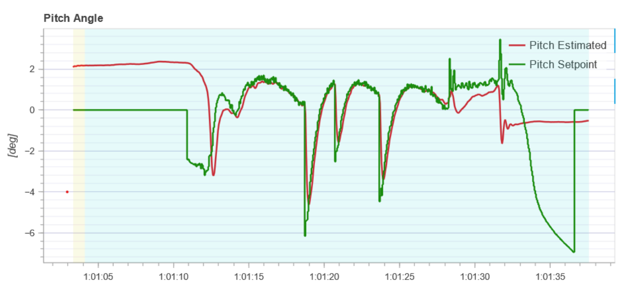

Figure 14: Pitch angle from pitch translational Flight

Figure 15: Pitch Angular Rate from pitch translational Flight

Figure 16:Local Position X from pitch translational Flight

Figure 17: Local Position Z from Pitch translational flight

ACKNOWLEDGMENTS

We would like to thank Christina Groenenboom, Susan Adams, Donna Fremont, the ERAU Physics Department, the ERAU Mechanical Engineering Department, ERAU Aerospace Engineering Department, and the Office of Undergraduate Research for their assistance in our grant management and guidance.

We would also like to thank John Rose for the wit and positivity he always provided in the early morning hours of our research.

REFERENCES

[1] G. Beni, "The concept of cellular robotic system," *Proceedings IEEE International Symposium on Intelligent Control 1988*, Aug. 1988. doi:10.1109/isic.1988.65405

[2] A. R. Cheraghi, "Past, present, and future of Swarm Robotics - arxiv.org," Arxiv, https://arxiv.org/pdf/2101.00671.pdf (accessed Dec. 12, 2023).

[3] G. Balamurugan and J. Valarmathi, *Survey on UAV Navigation in GPS Denied Environments* (2016).

[4]G. De Croon and C. De Wagter, "Challenges of Autonomous Flight in Indoor Environments," in IEEE International Conference on Intelligent Robots and Systems, pp. 1003–1009, Institute of Electrical and Electronics Engineers Inc. (2018) [doi:10.1109/IROS.2018.8593704].