Week 3: Mover_Node, and Flying Quadrotor

Individual Lab Report #3

Job Bedford

Team C: Column Robotics
Eric Sjoberg, Rohan Thakker, Cole Gulino

ILR3
10/29/15

Individual Progress:

My main responsibility for this past week was the completion of the mover_node for our AR.Drone2 software architecture. I also contributed to the creation of the second systems presentation and conceptual design of the power distribution board for task 12.

Mover Node:

As mentioned in the previous ILR, we are using the AR.drone, Figure 1, as test bed for our high-level code and position tracking algorithms. Ardrone_autonomy is a ROS library that enables us to control the drone over wireless communication.

Figure 1: Team C's AR.Drone2

The AR.Drone receives two main msgs type. One, the geometry::Twist 'cmd_vel', which orchestrate the drones velocity and angular velocities in 3 Euclidean space (X,Y,and Z). The second is the Std_msgs::Empty, which , when posted to particular topic, commands the drone to takeoff, land or reset. The mover node establishes publishers for each of the concerning topics, as seen in Figure 2. Then the node opens a tele-operated function, which takes user inputs via terminal key presses to control the drone like an RC car. The tele-operation is continuous and constantly pulls for another user command. Otherwise it continuously sends last given command.

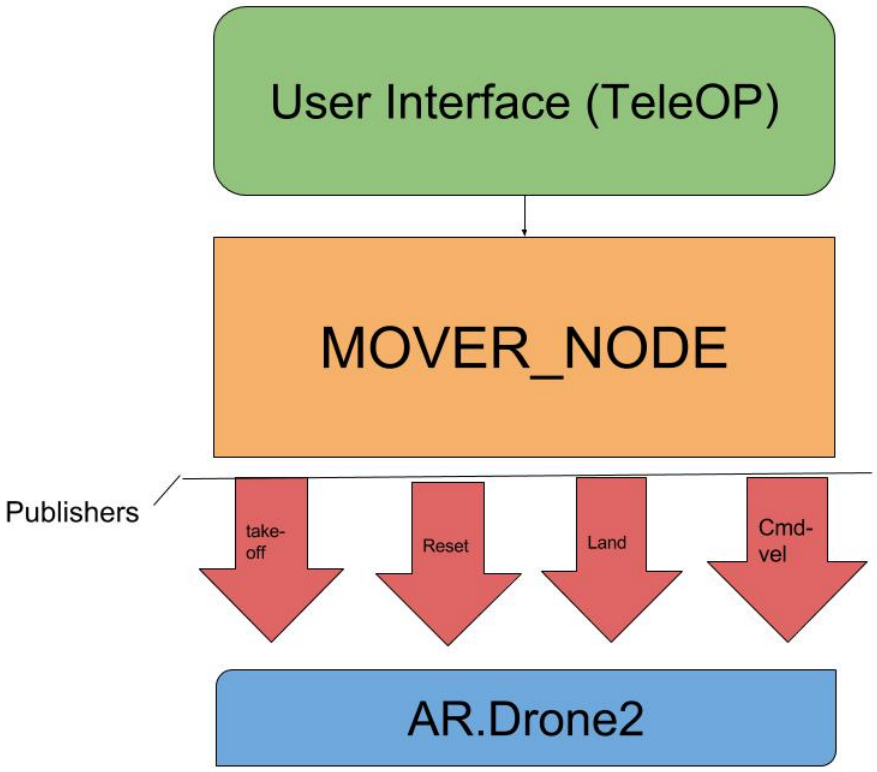

Figure 2: Mover_Node Diagram

A majority of the mover node code was scraped form an exist AR.drone controlling framework online. That framework derives from the 'TurtleSim' tele-operated node that is given in the ROS tutorials. I dissected and modified the existing framework for my own purposes.

Challenges:

The greatest challenge this past week was completion of task 7, the ROS familiarization assignment. Learning C++, ssh-ing into another computer via Launch file, and connecting to a server with incredibly slow communication were not easily surmountable tasks. Cole and I struggled with understanding how to set up the environment variables. We attempted using a specialized SSH server program that would remove the need for a password. The program backfired on us. My computer's connection to the 'rosassignmentWiFi' router was incredibly slow, taking 10 seconds or more to refresh the camera feed.

A lesser challenge was becoming accustomed to controlling the drone with the mover node. The AR.Drone has considerable variability in it take off. The drone velocity controls of the drone are also sensitive. The currently mover_node is using only a fifth of the drones full speed.

Teamwork

This week Erik work on a 2D mapping representation for the AR.Drone position-tracking objective. He also researched and documented ROS TF / odometry usage.

This week Rohan and Cole downloaded and compiled the Pixhawk source code to be later used on the IRIS. They also began work on a Simulation environment for the IRIS+ drone as well as began studying optical flow using the Lucas-Kanade Algorithem.

Upcoming Week

This upcoming week, the preliminary designs for the first dock Prototype will be completed. The finalized dock prototype will be presented in the fall demo. I will also be integrating the mover_node into the software architecture with my teammates.

In light of the set back due to the IRIS+ drone needing to be reordered, Rohan and Cole will be continuing development of the Quadrotor simulation in Gazebo as well as researching more on the 'pixhawk' source code. Eric will be continuing development on position tracking using the AR.Drone as well as researching ROS visual odometry.