Sensors and Motors Lab

Rohan Thakker

Team C: Column Robotics

Teammates: Job Bedford, Cole Gulino and Erik Sjoberg

ILR04

Nov. 13, 2015

1) Individual Progress

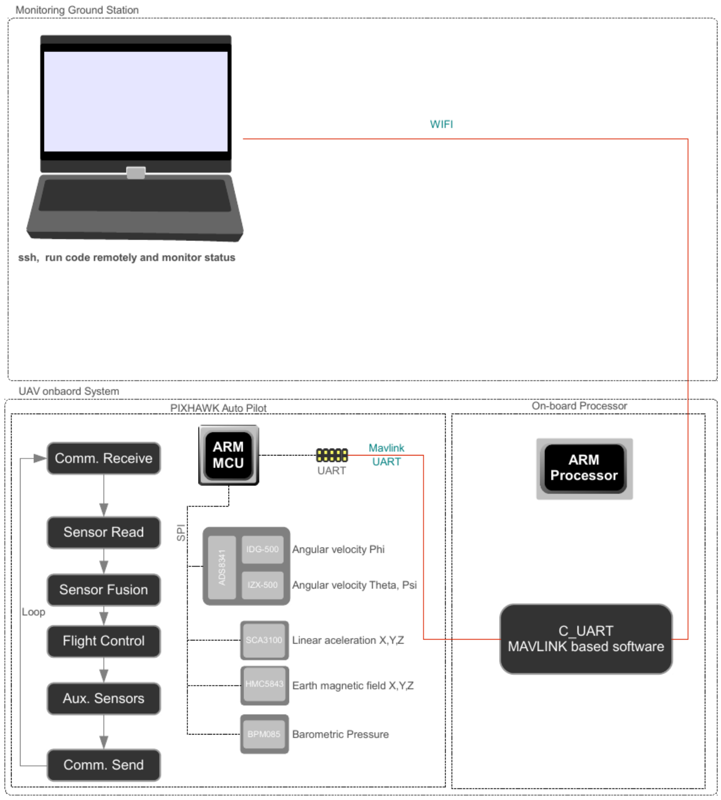

Figure 1 Cyber-Physical Architecture for communicating to Pixhawk source: [1]

I started this sprint by working on setup of the communication interface between the PIXHAWK and ROS on a PC. As shown in Figure 1, on-board processor (ODroid) communicates to the PIXHAWK using MAVROS protocol over UART. Communication between monitoring ground station (a remote PC) and on-board processor is achieved using ROS over Wifi. Job worked on setting up the communication between the PC and on-board computer; I worked on setting up the communication between the PIXHAWK and on-board computer. To make sure that this suits our requirement, I had to study the control algorithm that we will be using on the IRIS+.

As explained in the previous ILR, the quadrotor is a differentially flat system. Thus, the entire dynamics of the quadrotor can be represented in terms of a flat output space (y = [x,y,z,yaw]). As shown in figure 2, we have two cascaded loops running simultaneously. The inner loop is doing a close loop control on the attitude (roll, pitch and yaw) and the outer loop is doing closed loop control on the position. We will be running the outer loop on the on-board processor (at about 60 Hz) and inner loop on the PIXHAWK at (250 Hz). The UART communication supports a baudrate of 921600 bps. Hence we should have enough bandwidth to run our outer control loop at the desired frequency. After deciding the control strategy, I followed instructions [3] to set up the communication interface.

I also researched various algorithms for visual odometry. I originially started by implementing Lucas-Kanade based optical flow algorithm using opencv. But noticed that the

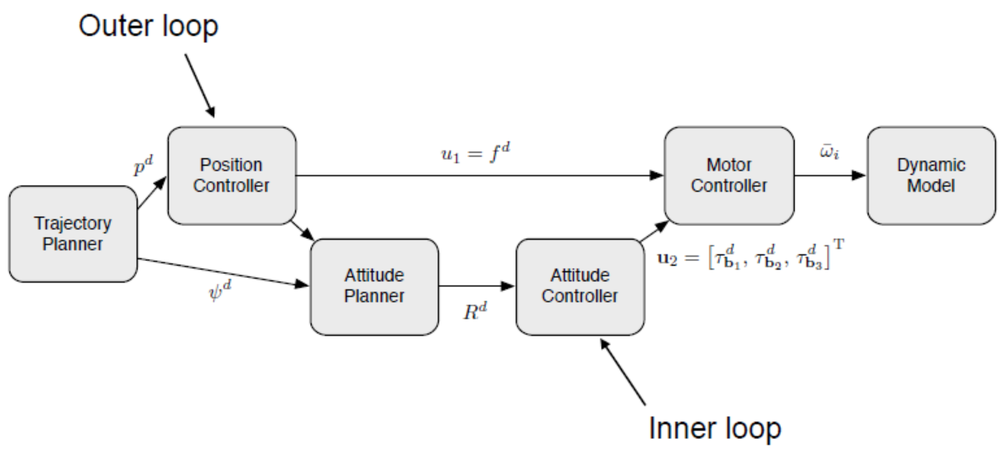

Figure 2: Controller for Quadrotor source: [2]

drifts kept increasing significantly over time because I was integrating the velocity estimates over time. After researching, I discovered that most quadrotors use PTAM (Parallel Tracking and Mapping) based algorithms. The algorithm runs two threads of tracking and mapping in parallel. The mapping thread creates a 3-D map by triangulation using SIFT/SURF based feature point correspondences. The tracking thread first creates an intial estimate of the its pose. It uses this estimated pose to identify the points in the 3-D map that are expected to be visible. Finally, localization is done by calculating correspondance between 3-D point in the map and their location in the image. The key idea is to maintain a map of 3-D feature points, so that we can use triangulation to correct the drift.

There have been many algorithms that have been built over PTAM. Semi-direct Visual Odometery (SVO) is one such popular algorithm used for quadrotors. PTAM algorithm is computationaly intensive and connot run on-board. This is because calculating the SIFT/SURF based feature points is very expensive. Hence, instead of calculating feaure points using SIFT/SURF the SVO algorithm identifies feature points directly on their intensity values. This idea is also used in other popular algorithms like LSD-SLAM.

Since the IRIS+ arrived sooner than expecited, so I decided not to waste time on simulation and directly work on the actual hardware.

2) Challenges

To set up the communication interface, I was using the USB port on the PIXHAWK. However, I discovered that it does not support a baudrate of more than 56700 bps. Hence, in order to meet our requirements for the frequency of the control loop, we have to shift to UART, which supports a baudrate of 921600 bps.

After arrival of the IRIS+, on our first flight, Job and I, accidentally crashed the drone a couple of times. Due to this, all the 4 propellers were deformed. I ordered propeller guards that fit to the bottom of each BLDC motor. I also ordered many extra propellers, battery, battery charger and PX4FLOW camera. The PX4FLOW camera provided by 3DR gives visual odometry updates. I also confirmed with the previous year's team that this works well. Hence, this is a good risk mitigation strategy if we are not able to get visual odometry working on sony playstation eye camera for the IRIS+.

We were using the ardrone_autonomy package to control the ARDRONE. This package was estimating the position directly by integration of visual odometry. Erik ran a test flight for 10mins and the drift was as bad as 30m. Hence we shifted to another package tum_ardrone which fuses the data using an extended kalman filter. Further, this package also has a provision for PTAM based visual odometry. The result of this was shown during the demo.

On testing the SVO algorithm [4] using sony playstation camera, I discovered that the algorithm works well for small displacements. But, for any large displacements or relatively small rotations, the robot loses the feature points in the map. Once this happens, the algorithm is not able to re-localize itself. Hence, to make this work better, I need to better understand the various parameters and tune them. We have already achieve good state estimation on ARDRONE, i.e. required for our Fall Validation Experiment. Hence, I will resume this work when we are trying to get visual odometry working on IRIS+.

3) Teamwork

Table 1: Teamwork

Job Bedford	**Cole Gulino**	**Erik Sjoberg**	**Rohan Thakker**
Decide mounting location for sensors and SBC on IRIS+	Designed power distribution board	Evaluated performance of odometry using ardrone_autonomy	Set up communication between on-board computer and PIXHAWK
Set up communication interface between ground PC and on-board computer	Evaluated performance of odometry using ardrone_autonomy	Evaluated performance of tum_ardrone using EKF and PTAM	Studied visual odometry algorithms and tested SVO
	Research navigation stack in ROS	Research navigation stack in ROS	

The Table 1 shows the tasks done by each of our team members.

4) Plan

Since we have completed most of the work required to be done on the ARDRONE for the demo. Erik will be focusing on performing extensive testing to make sure the results are repeatable. Cole will be working on refining the pipeline of our navigation stack.

Job will be leading the design and fabrication of the docking mechanism with support from the entire team. We aim to complete the fabrication the first prototype of the docking mechanism. I will be designing and fabricating the parts required to mount the single board computer and the camera on the IRIS+. I will also be spending time on researching state estimation and control methods for quadrotors.

5) References

[1] https://pixhawk.org/dev/offboard_control

[2] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robot. Autom. Mag., 19(3):20–32, Sept. 2012.

[3] http://wiki.ros.org/mavros

[4] https://github.com/tum-vision