Visual Odometry:

Current Tasks:

We are still trying to get the raspberry pi to connect to the pixhawk. Also our SD card became corrupted and we lost our files. So at the moment we are also recreating our previous work.

Zavia is working on recreating our previous work.

Trent and Quinton are working on getting the connection between the pixhawk and raspberry pi to work.

Dead reckoning Team:

Current Tasks:

Rich: Look at existing IMU code's Bias control and see if it is sufficient enough for our data outputs. Looking at it, from a few of our tests, it only takes out about 20-30% of the total bias on the IMU. The timestamp to detect the bias was 0.05 seconds, changing that to 5 seconds to detect and average bias, the code takes out about 50-60% of the bias.

Brendan: Working on implementing Kalman Filter from Cube Sat project into existing IMU code and troubleshooting.

Ryan: Working on troubleshooting the running kalman filter. Knowledge from the previous Cube Sat project will help us develop the kalman filter further. I will try and work on the bias calibration code this weekend by taking static tests for the Matlab code.

Pi Integration/GPS:

Completed Tasks:

Nick: On friday we found strong evidence that Dronekit will help ease our situation and that MavLink will work for sending GPS signals.

Moving forward I will work on increasing my understanding of dronekit and passing that info on to the team.

Current Tasks:

Josh: Preparations are being made to test the GPS data output from the Raspberry Pi to the PixHawk. An adapter was made to connect the GPIO pins of the Pi to the GPS1 port of the PixHawk. A dummy script will be written to output an NMEA formatted GPS signal to test readability in Ardupilot. Once the script is confirmed to be readable and optimized, a program will be written to gather NMEA data from the mounted GPS and modify it to be output to the PixHawk as spoofed GPS data. An online Python compiler was found that can be used for testing script without using the Raspberry Pi. https://www.onlinegdb.com/online_python_compiler

The connection between the Raspberry Pi and the PixHawk is undergoing troubleshooting. An error shows that an initiation file is not found in the root folder, which is a common error observed in existing setups. More research is being done on solutions involving the ports used to connect the two systems. It was discovered that the PixHawk we used was faulty, and a new unit enabled a connection.

Rushal: Finished working on communicating Pixhawk to Raspberry pi. When editing boot/comdline, make sure root=/dev/mmcblk0p7, make sure mmcblk0p2 is changed to mmcblkop7. Also, when pixhawk is connected to Pi, disconnect the pixhawk usb connection to computer. When typing mavproxy**.**py --master=/dev/**Serial0** --baudrate 921600 --aircraft MyCopter, make sure to use Serial0 and not ttyAMA0. Keep default badurate that is provided through mission planner. Pixhawk without case is faulty. One with black case works, pixhawk on drone works.

Vivian: Pi has been successfully connected to Pixhawk. The GPS subgroup worked more with the Vis-Od group as we shared this task goal. Next we will work to have the Pi extract longitude/latitude NMEA code from the GPS and retain this as the initial GPS lock to build off of. We are going to do more research on how to make an initial GPS lock log, have the pi retain this log then command shutoff GPS. The DroneKit API (application programming interface) will then use the Pi's spoof GPS signals to give flight commands to the Pixhawk. https://dev.px4.io/v1.8.2/en/robotics/dronekit.html

Nick: This week I set up DroneKit and MAVlink on the Pi. MavLink is an opensource library of low-level code used to automate the piloting controls of a drone (Hovering, moving, and any minor balancing adjust minutes). Dronekit is a companion library used to send MavLink commands in a simple way that is through upper-level dictations. I have spent the rest of the week gathering sample codes for DroneKit to better understand how to use it. With this I also reorginized our teams tutorials and added one for setting up Dronekit & a basic simulator on an indivuals PC. I set up doccuments to start sharing information learned on dronekit and python coding, and perhaps most importantly I setup a Git repository accompanied with an explanation of what Git is and why I think this project should pick it up.

Python and Dronekit Intro | Drone Programming (2019)
http://ozzmaker.com/forums/topic/is-it-possible-to-turn-off-power-to-the-gps-independently-from-the-imu/
http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html

Future Tasks:

I would recommend we find out if

a. GPS Can be sent through telecom
b. If GPS can be used through the GPS port using MAVLink & DronKit