Questions for client

Preferred way to transfer data from the sensors to the visualization? (We are leaning towards wifi or bluetooth)

* Bluetooth from Raspberry Pi to a mobile app
* Wifi from Raspberry Pi to a mobile app
* Physically transfer a USB from the Raspberry Pi to a laptop

Preferred device for visualization? (We are leaning towards mobile app)

* Visualize on a mobile app – this would require bluetooth or wifi for data transfer
* Visualize on a laptop – this would probably require data transfer by USB

**Proposed milestones**

Visualization/App

* Basic mechanics by 11/8 (logging in, navigations between screens, a basic 3D view)
* Working 3D visualization of a plane with some dummy data by 11/22
* Visualization based on data collected through bluetooth, add in airspeed and pilot’s inputs by 12/6

Hardware

* Able to extract data from AHRS/INS sensors into the Raspberry Pi by 11/15
* Prototype for pilot input/cable tracking by 11/22
* Prototype for transferring all data to the visualization by 12/6

Send docs in .docx format in the future

Start by using an SD card for data transfer

* Or USB cable

Dropping a micro sd card in the plane is a big deal - would have to find it no matter what

* Might not be as bad if it’s just dropped in the cockpit

Could export data to kml file - XML for geographic stuff (GIS)

* Points, altitude, time,

Look into Xplane – takes in data, does some kind of simulation

* It is a full featured flight simulator
* Free
* Receives data from a plugin, which can receive a UDP stream

A mobile interface is not necessary, also, it would be nice to have a larger screen

FAA requirements allow attaching stuff with zip ties to the struts that connect the wings to the fuselage

Can also run a cable outside the access holes at the bottom of the plane, but there’s not much to attach it to (not allowed to make permanent modifications)

Also, landing gear could be used to attach stuff

20hz is low for inertial data, bc engine spins at 40-50Hz – aliasing is an issue

GPS and control stick is fine at 5 or 20 hz

For raw data, want several thousand hertz

Keeping the original raw data will aid in smoothing the data (AHRS presents pre-smoothed data)

Keeping raw data would not take up too much storage, size wouldn’t matter to the user – only get rid of raw data if it’s causing a storage issue

GPS – get more accurate position by trading off collection frequency

* Also, there’s a ~20-50 ms delay, must account for this to connect it to inertial measurements
* Use post processing smoothing

As a default, keep all the data

May record things asynchronously, don’t record data when data doesn’t exist. E.g. don’t write zeros when a sensor isn’t collecting data frequently enough

Also, never write redundant data

Don’t change timestamps on data to align it

Make plan B milestones for if hardware shipping is significantly delayed

Buy redundant parts so the project is not dependent on every single piece of hardware not failing

Ken thinks this will all take 100-200 (closer to 100) engineer hours

Note: allocate hours to each subtask

Deadlines will probably require 10 hours per week

Start weekly standups

* Ask each person 3 things: What they did last week, what they will do next week, and any blockers

Ad-hoc assignment of work will result in one person doing all the work, and this will also make it harder for other people to join in and help them

Actionables:

* Make small changes to the PDRR, send to the client, and receive a signature. Also, could maybe email the updated version to Prof. Pisano for completeness
  + Pretty much all the updates have been made in “Copy of PDRR Final” document
* Create a second version of the milestones (“Plan B”) in case we have trouble receiving hardware on time, and it delays our progress on hardware milestones
  + Send both plans to the client
* Organize tasks much more thoroughly: Assign an expected number of hours to each subtask, take account of each person’s strengths, and assign specific tasks to each team member
  + Use the Notion for subtasks, assignments, and deadlines
* Start weekly standup meetings