Fly Sense

Team C – ILR05

22th November 2017

Joao Fonseca Reis

Shivang Bhaveja

Harikrishnan Suresh

Nick Crispie

Sai Nihar Tadichetty

Work done this week

In the last two weeks we have done a ton of productive work!

The different work packages are slowly but surely falling into place, and with an extra push during the thanks giving break we will get there for FVE.

UI/UX

- Substantial progress was done on the graphical user interface, with the home screen, HUD (telemetry) and bird's eye view (surrounding obstacles) deployed and integrated with the Jetson (we actually finished this literally 5 minutes before the our PR4 – check section on problems faced this week 😊!)

- We did our first voice command control using Google API's given that the AR Epson-BT 300 uses the Android operating system
    - A very good insight was that numbers and letters (in English) are difficult to distinguish, but it turns out that "computer", "alpha", "bravo" and "Charlie" are very easy to detect… so we changed the different voice commands to these!
    - It worked very well in Nihar's Android phone, but the Epson does not have a reliable internet connection and the Google APIs are blocked (we only found this one later researching the Internet)… so we deployed an offline voice command control using the library "pocket sphinx"

- The sound warnings code is also ready to integrate with the Epson. We went through several designs in order to ensure that:
    - We could use a process capable of handling multiple inputs real time
    - That warnings escalated gradually
    - That warnings did not overlap (second warning only issued when first is finished)

Sensing and mapping

- We got our quadcopter out of the lab for a spin (mounted on a cart in line with our FVE experiment) a few times and managed to connect the lidar, the Jetson and the IMU together for the first time
- We also managed to connect the quadcopter telemetry with the Jetson and push a real-time monochromatic image of obstacles to the Jetson
- We managed to process the first datasets from NEA (conversion was needed from double to float to visualize in rVIZ and high decay time allowed to visualize a beautiful map for the first time, given that NEA is rotating the Velodyne and that is the only way to extra information out of the bag files)
- Good progress was also made on mounting the Power Distribution board

Problems Faced this week

This week during the PR4 a very funny incident took place (funny in hindsight). We got the first obstacle image pushed to the Epson literally 5 minutes before our status update.

Then Nihar connected its phone to the Epson to do some testing and the entire system collapsed when professor Dolan, Ricky and Yoga finished the status update with Group D. Nihar was going to present our status update and I told him not to worry, that I would reboot the system while he presented the rest.

Nihar presented and gave the floor to the different demos being shown (systems integrated, data from NEA, voice commands online, …). I managed to reboot the Epson on the fifth attempt, in the middle of my demo.

- This demo started describing how we would change the frequency of the sound warnings
- Evolved to an example using Nihar's Android phone (a simple beep sound that can be played with more or less time between beeps, with different volumes for left and right year, looped or even played at different speeds…
- … and finalized with a live demo of the old voice commands using the Google APIs in the Android phone on Nihar

I initiated the demo of the voice commands and it was working well, but the fact that group D was talking in the background was confusing the app. We told them about the need to remain silent and suddenly we started having a lot of network and server down errors (that we had never had before….).

We later found what had happened…. Nihar had "wired" his phone to automatically connect with the Epson (and in the process disconnect itself from the Internet) when the Epson was back online…. This was the test he was doing when the entire thing collapsed 1 minute before the status update… but he did not know that it would reconnect itself again!!!!

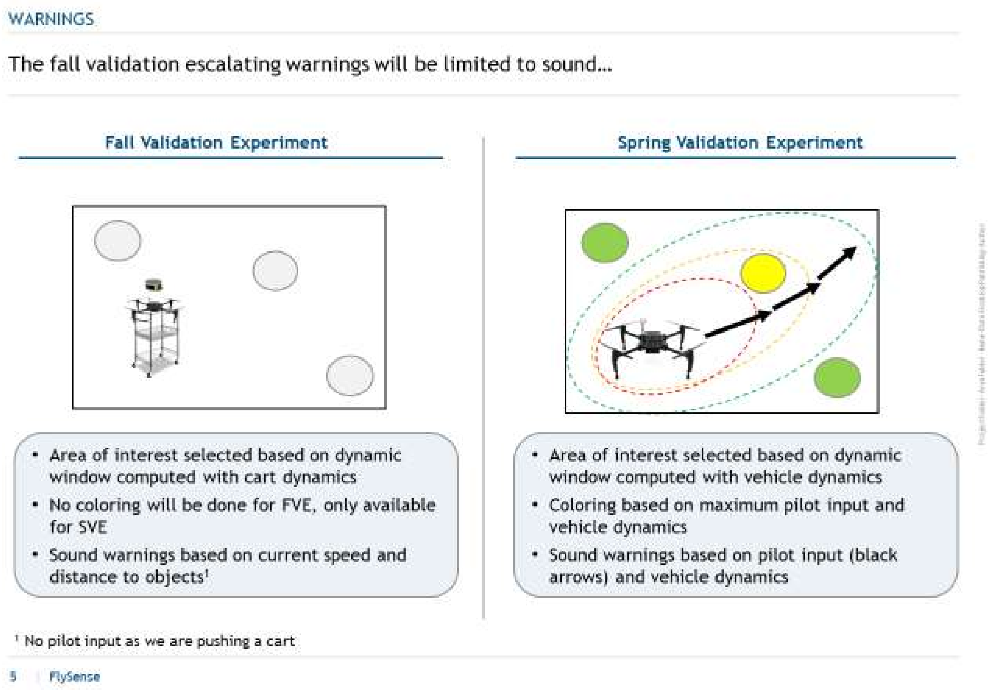
In the end, the voice commands demo using the pocket sphynx library was a success, but I did not need the multiple heart-attacks I had half-way through the PR4.

Individual achievements for the past 2 weeks

This week I have: written the code for

a) Generating sound warnings (Java Android)
b) Written the code for manipulating the escalation of the sound warnings as a function of the time to impact in the nearest object (which will have as an input the objects coming from the mapping sensors)

I also helped Nihar integrating the Interpreting basic voice commands (Java Android) using the Google APIs and in trying to figure out why it worked on his phone but not on the Epson.



WARNINGS

The fall validation escalating warnings will be limited to sound…

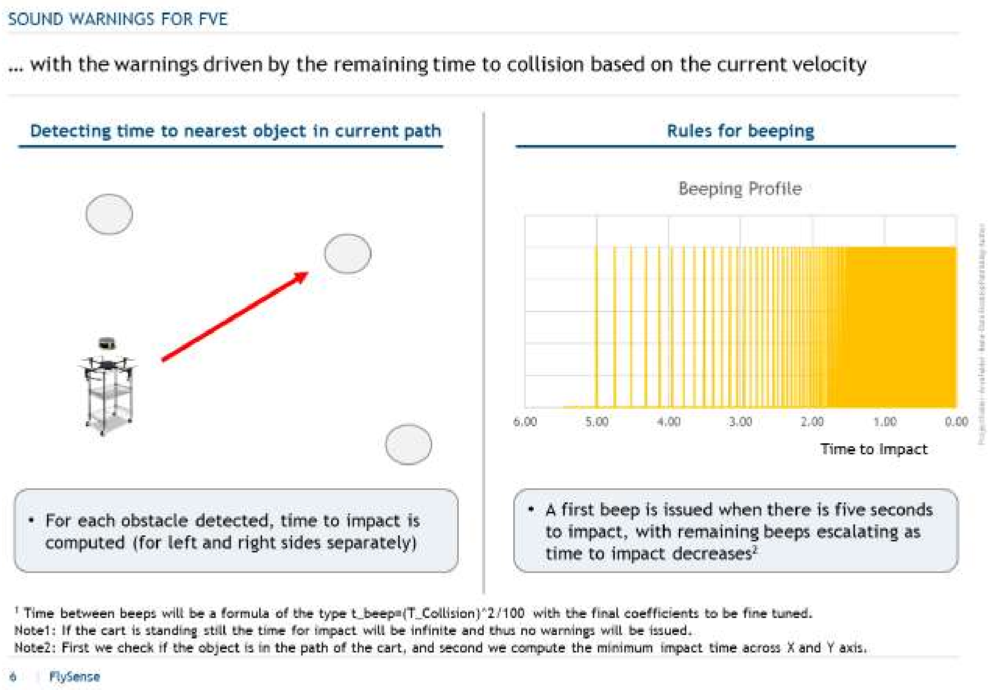
Fall Validation Experiment

- Area of interest selected based on dynamic window computed with cart dynamics
- No coloring will be done for FVE, only available for SVE
- Sound warnings based on current speed and distance to objects[1]

Spring Validation Experiment

- Area of interest selected based on dynamic window computed with vehicle dynamics
- Coloring based on maximum pilot input and vehicle dynamics
- Sound warnings based on pilot input (black arrows) and vehicle dynamics

[1] No pilot input as we are pushing a cart



SOUND WARNINGS FOR FVE

… with the warnings driven by the remaining time to collision based on the current velocity

Detecting time to nearest object in current path

- For each obstacle detected, time to impact is computed (for left and right sides separately)

Rules for beeping

Beeping Profile
6.00
5.00
4.00
3.00
2.00
1.00
0.00
Time to Impact

- A first beep is issued when there is five seconds to impact, with remaining beeps escalating as time to impact decreases[2]

[1] Time between beeps will be a formula of the type t_beep=(T_Collision)^2/100 with the final coefficients to be fine tuned.
Note1: If the cart is standing still the time for impact will be infinite and thus no warnings will be issued.
Note2: First we check if the object is in the path of the cart, and second we compute the minimum impact time across X and Y axis.

Milestones for next two weeks:

With Nihar:

a) Integrate the sound warnings generating code into the Epson
b) Fine tune pilots display and remove bugs (delays, unreliable output, …)

With Shivang/Hari:

a) Integrate the dynamic window code in the Jetson (to segment obstacles)
b) Integrate the sound escalation code in the Jetson
c) Understand how to fast track the mapping effort

With Nick:

a) Support in the preparation of the FVE infrastructure

With the entire team: Prepare the Fall Validation Experiment (the clock is ticking!!!)

Key risks:

We have a lot of work in the coming days on the ramp up to the FVE, with risks concentrated on:

a) Integration
b) Testing and debugging
c) Generating maps

The last point is the one I see as the biggest risk so far, as we have not yet found a way to generate minimally presentable maps in real time.

We can generate beautiful maps using NEA data sets (good GPS and IMU) with long decay times, but that would probably not work well real time (memory and processing power constraints).

Hari will be working remotely during the thanks giving break, and the rest of us will be in Pittsburgh trying to fuse all the remaining items together.

Fingers crossed for FVE! (and Encore....)