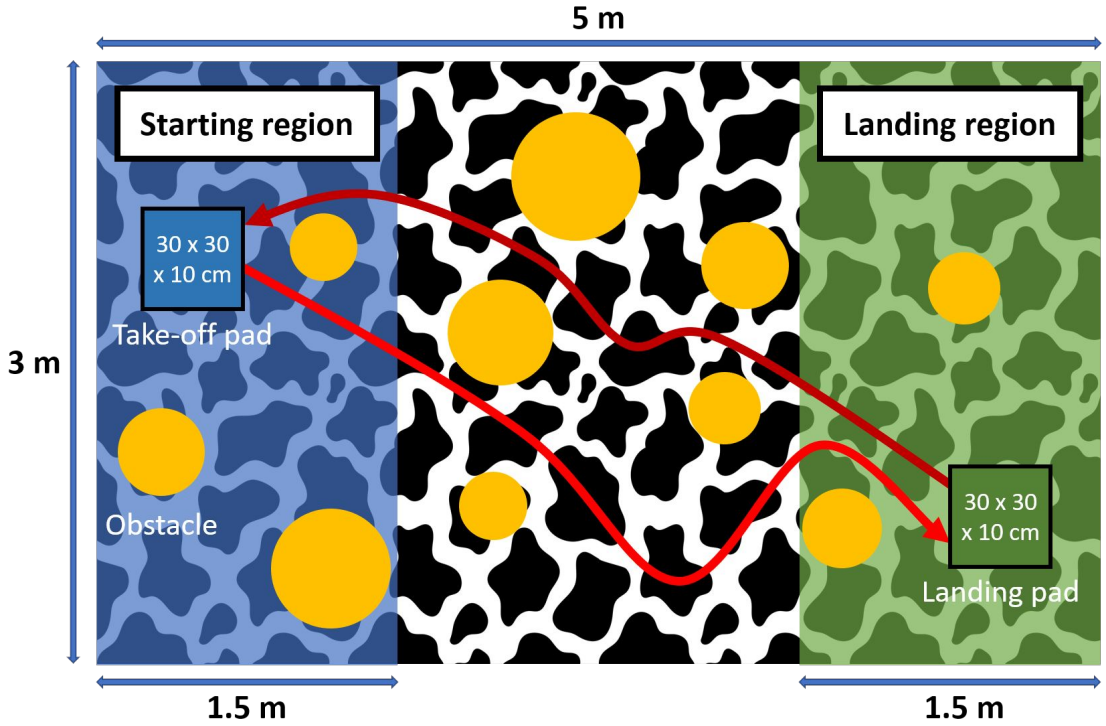
Crazy Practical

Introduction

Teaching assistants:
Shushuai Li, Mohammad Askari, Won Dong Shin, Valentin Wüest

Task Setup



- Arena: pattern tiles (3 x 5 m area)
- Platform: 2 x carton box (~ 30 x 30 x 10 cm^3 volume)
- Drone: initially placed on the take-off pad
- Task: navigation and obstacle avoidance
  - Estimate position from flow deck
  - Use multi-ranger to detect and avoid obstacles
  - Use z-ranger to detect and land on the platform

Last year’s exam

Grading

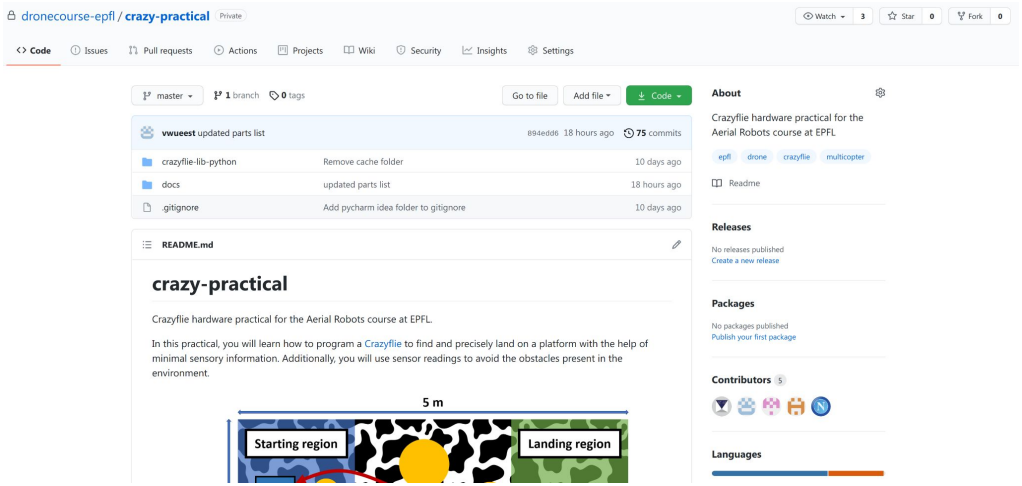
● **Level 0**: presentation and video of working algorithm	0-4	<=4
● **Level 1**: take-off, avoid obstacles in the way	0-0.5	<=4.5
● **Level 2**: locate and land on the landing pad	0-0.5	<=5
● **Level 3**: go back, land on take-off pad	0-0.5	<=5.5
● **Level 4**: finish Level 1-3 within 2 minutes	0-0.5	<=6

Group & hardware

- There will be 15 groups; each group will be made of 4 students
- Choose your group in Moodle before the next lecture (April 12th)
- During next lecture (April 12th), each group will be given a drone kit
- Return the drone on the date of the final exam (May 31st)

Github repository

- Contains the instructions



https://github.com/dronecourse-epfl/crazy-practical-2022

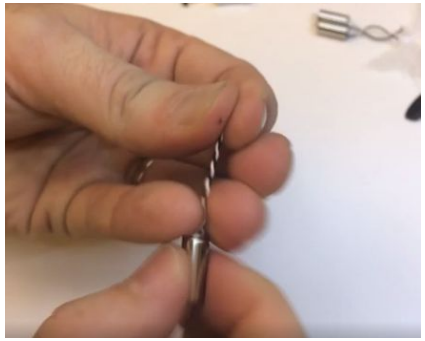
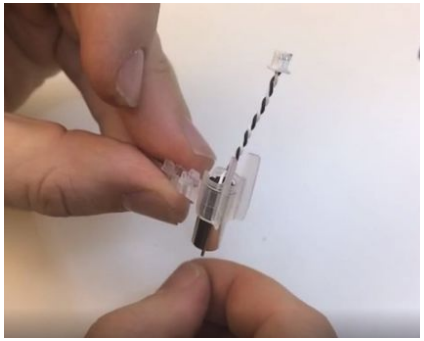
0. Privately fork the project

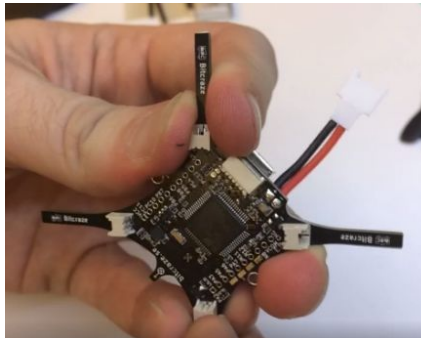
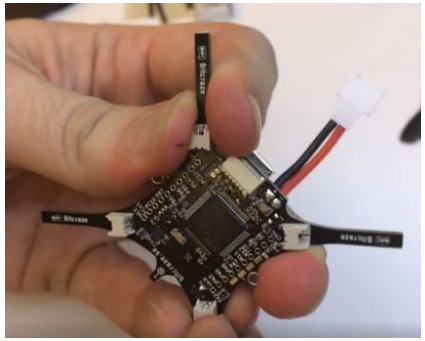
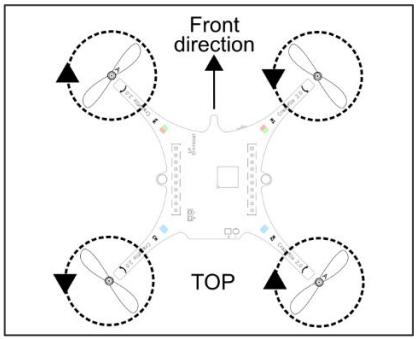
- Create a private fork of a public repository · GitHub
- Add your teammates as collaborators
- Send code, presentation and video of the task by 29th May, 23:59

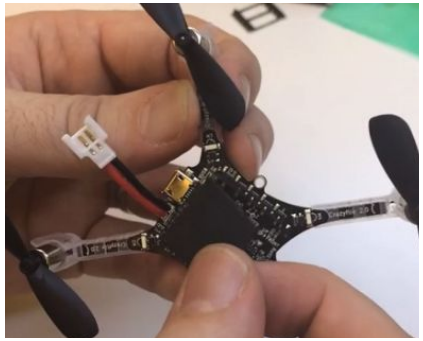
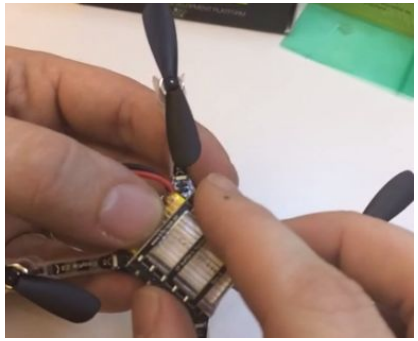
1. Unpacking

Visit the bitcraze documentation:
https://www.bitcraze.io/documentation/system/

2. Assembling

3. Installing the Python library

- **cflib: Crazyflie python library**
  - API written in Python
  - used to communicate with the Crazyflie
  - used to control the Crazyflie
  - used by cfclient, the Crazyflie PC client
- Visit the doc here:

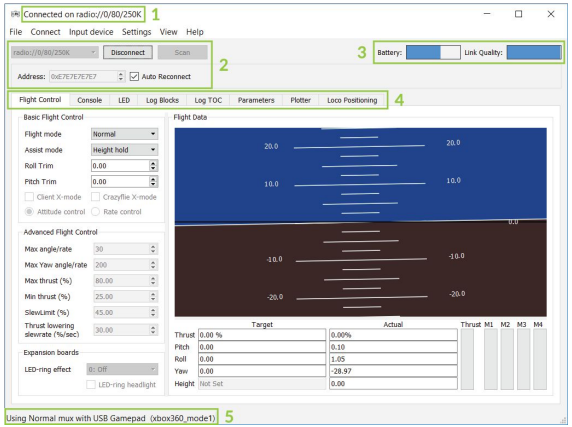
  https://www.bitcraze.io/documentation/repository/crazyflie-lib-python/master/

4. Setting up the radio interface

- Make sure you have the usb permission for the radio interface
- Configure the crazyradio and set your crazyflie address
- Remember to change the uri in the files of your algorithm

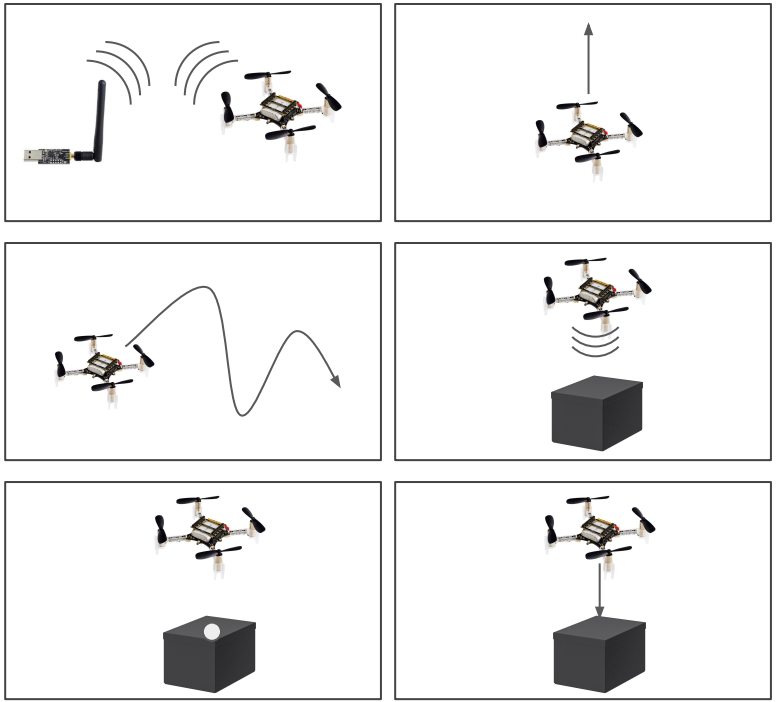
5. Installing the Crazyflie client

- **cfclient: Crazyflie PC Client**
  - written in Python
  - used to flash, change settings, control, and visualize logs of the Crazyflie
  - implements the user interface
- Visit the doc here:
  - https://www.bitcraze.io/documentation/repository/crazyflie-clients-python/master/
  - https://github.com/bitcraze/crazyflie-clients-python

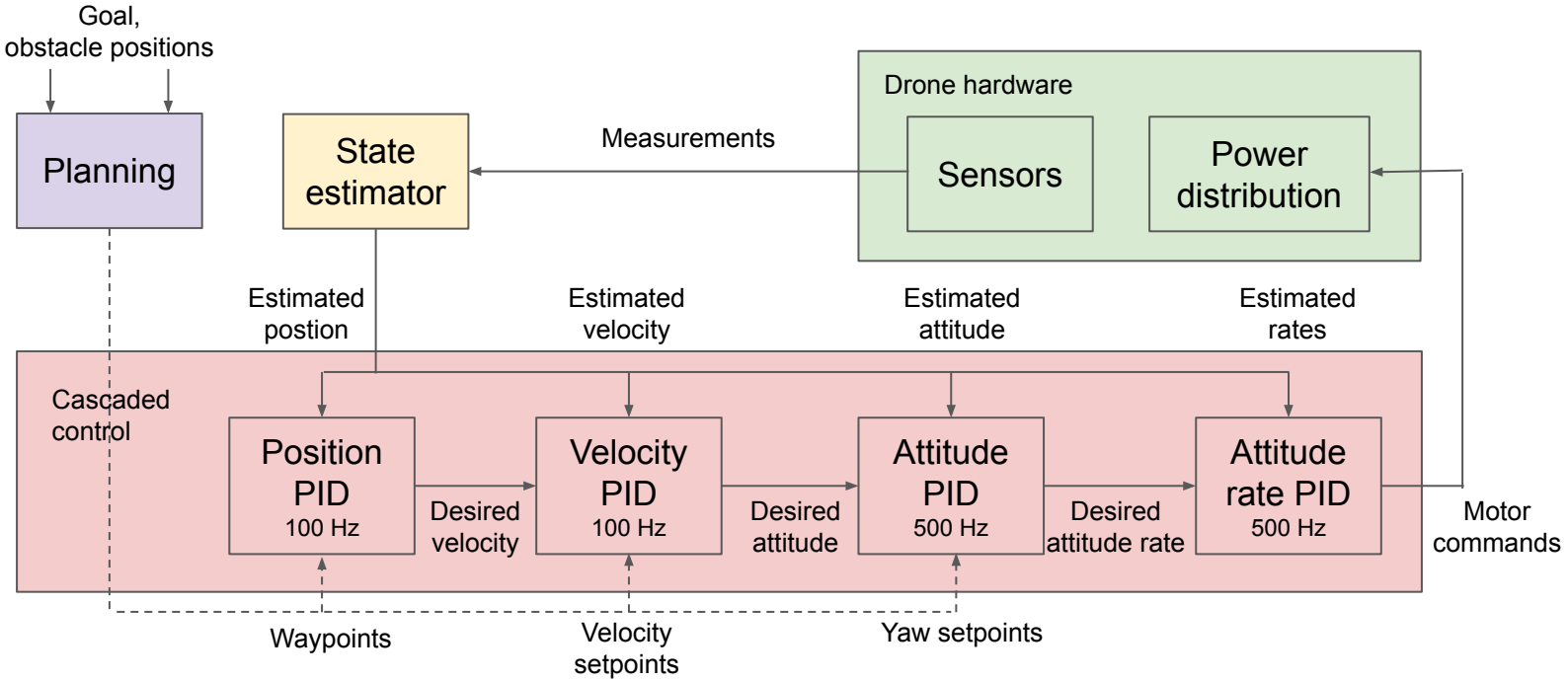


6. Coding your algorithm

- Connect to the crazyflie
- Take off
- Explore the arena
- Avoid collisions
- Detect the box
- Estimate the box center
- Land on it
- Go back to start point

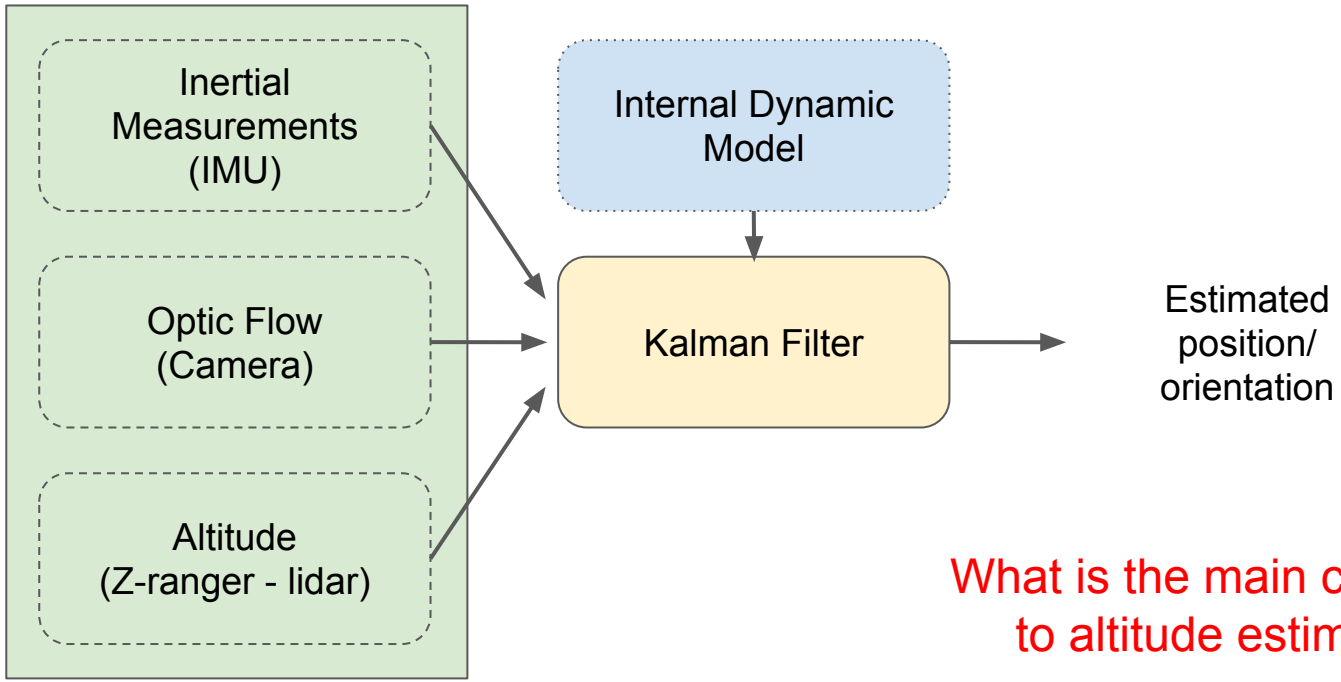


Crazyflie software architecture



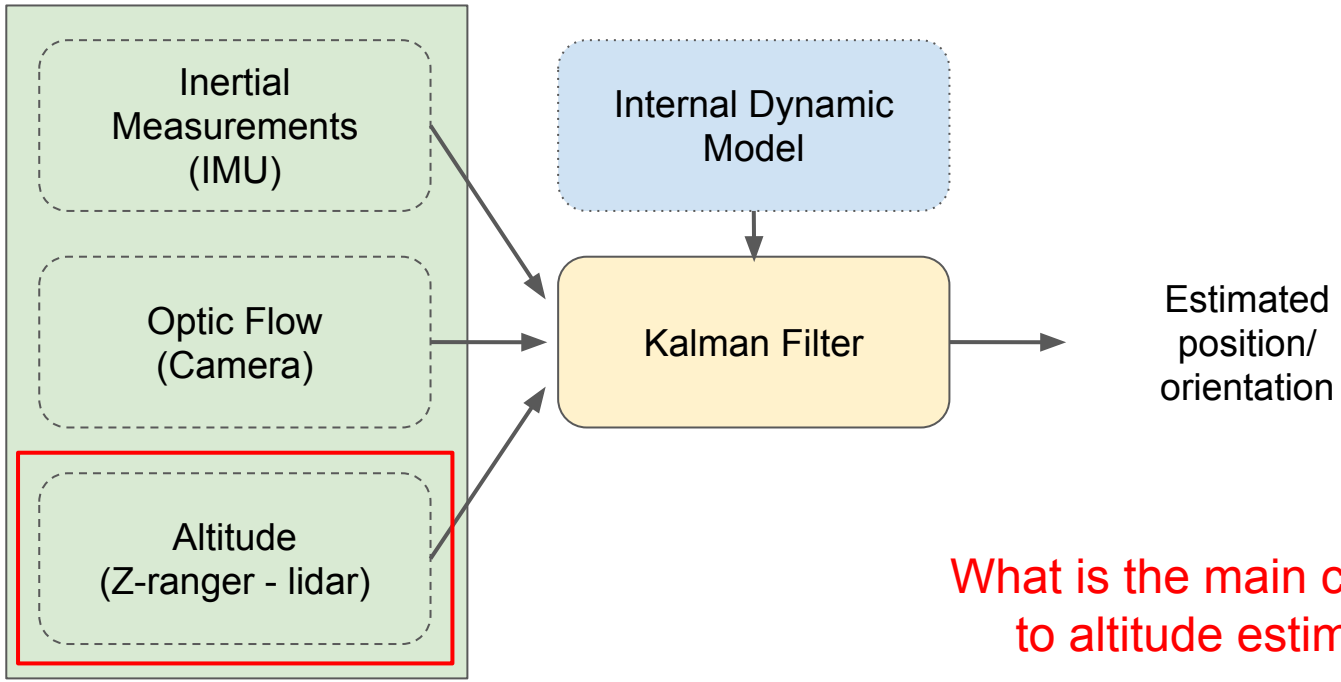
https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/sensor-to-control/controllers/

Altitude estimation



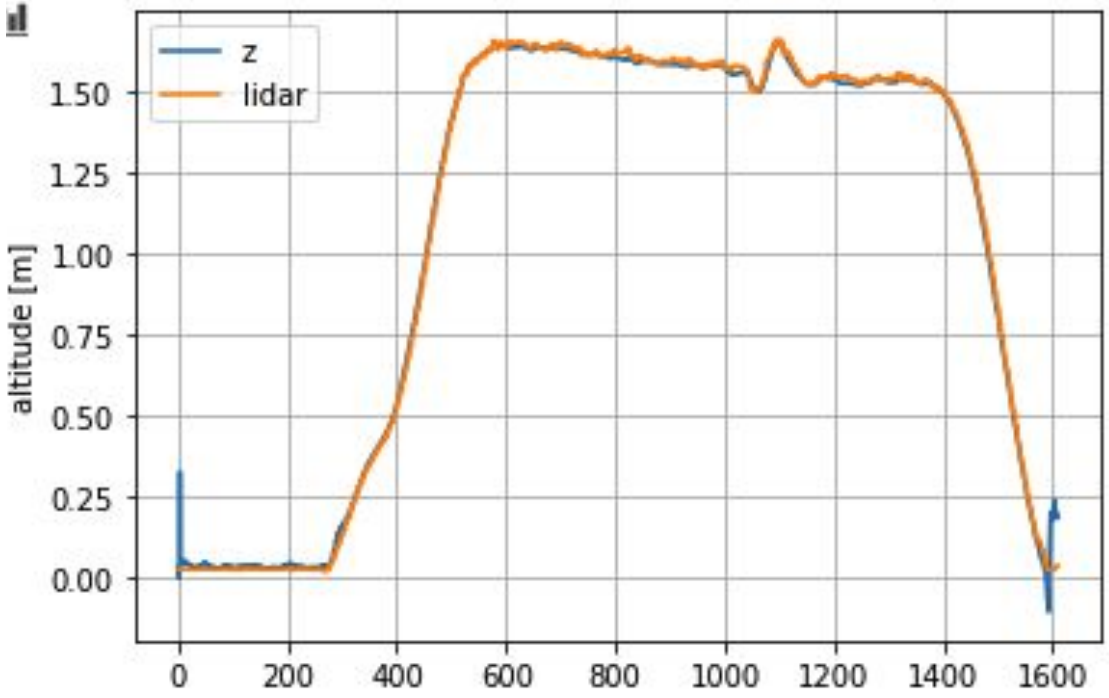
What is the main contributor to altitude estimation?

Altitude estimation



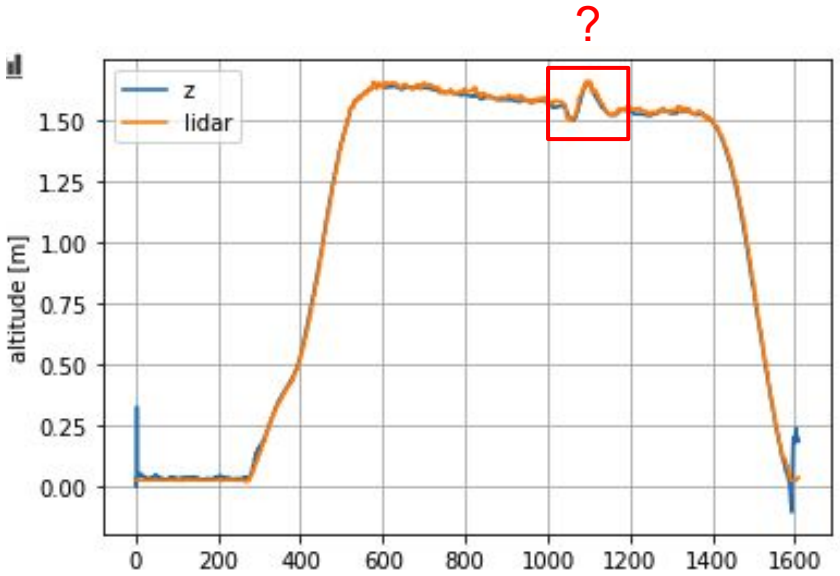
What is the main contributor to altitude estimation?

Forward flight at constant altitude



Edge detection

When flying forward and passing over an obstacle (the box, in our case)



What’s next?

- Do the installation
  - https://github.com/dronecourse-epfl/crazy-practical-2022
- Start coding
  - TAs will answer your questions

Test the drone and start coding (Apr 12)

- Sign the from to get your drone
- Change your drone's address with cfclient
  - Address is explained in https://github.com/dronecourse-epfl/crazy-practical-2022
  - Radio channel = 80 + 10*(mod(<team_number>,3) - 1); Crazyflie address = 0xE7E7E7E7<team_number>
- Change address in each baciclog.py and test your drone with that file
- Put drone **in the arena** when testing logandfly.py or position_commander.py
- Update your crazy-practical-2022 repository with git pull command.
- Every time before flying the drone, **restart it** to get good position estimation.
- TAs will answer your questions

Some notes

- Initial position of the drone will be given one day before the final exam
- Video details
  - Keep the drone in the field of view
  - Don’t edit the video (except for speeding up)
  - Display a timer at the beginning and the end of the flight
  - Record one successful flight in an obstacle-rich environment
- Commander documentation (such as landing and taking off)
  - https://www.bitcraze.io/documentation/repository/crazyflie-lib-python/master/api/cflib/positioning/position_hl_commander/