Study & design of the control architecture of an UAV-based additive manufacturing of architectural structures

Alexis PAQUES & Nabil NEHRI

UCL
Université catholique de Louvain

ECAM
Brussels Engineering School

08 July 2016

Specifications

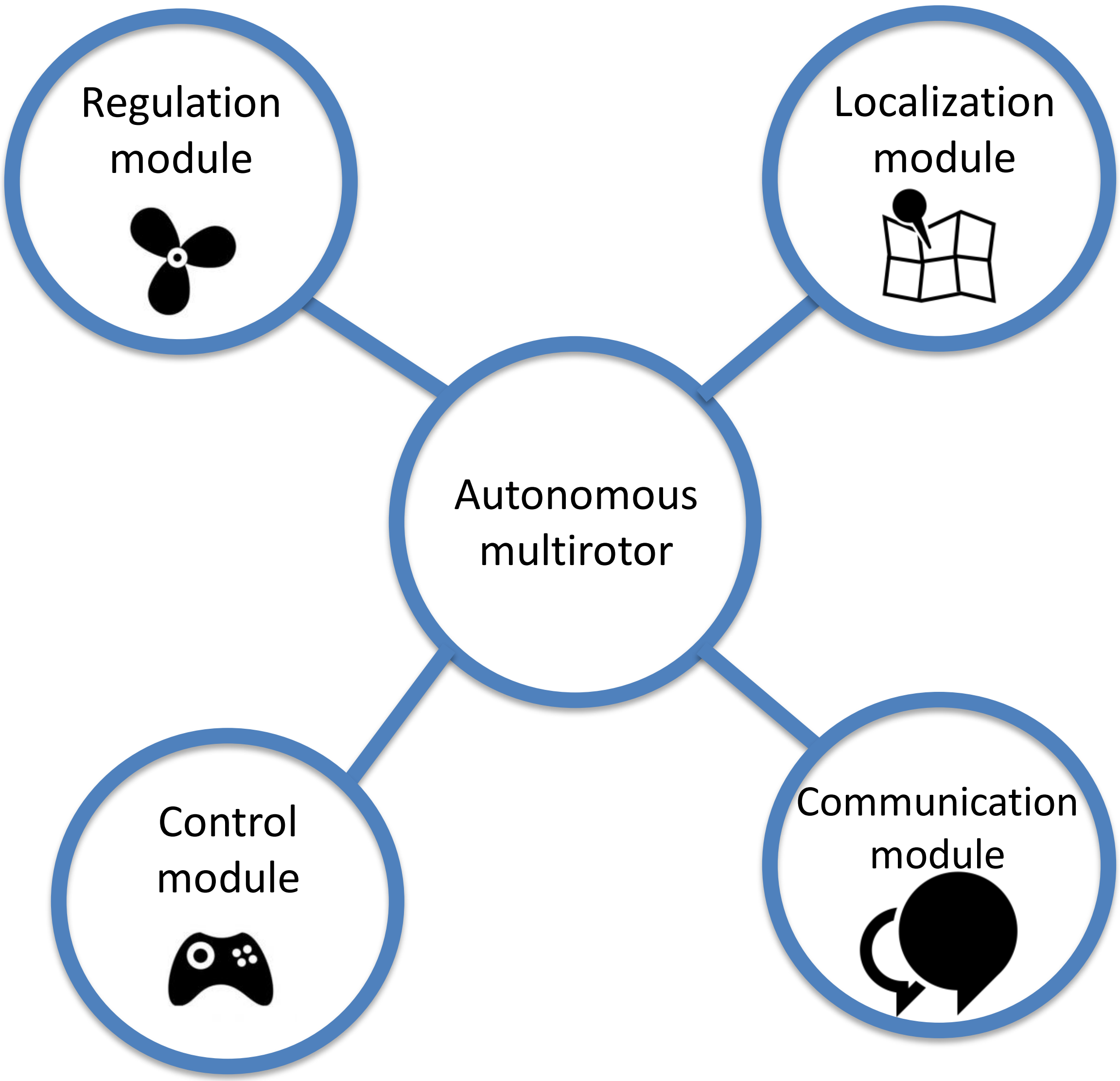
Localization system

Control system

Autonomous flight

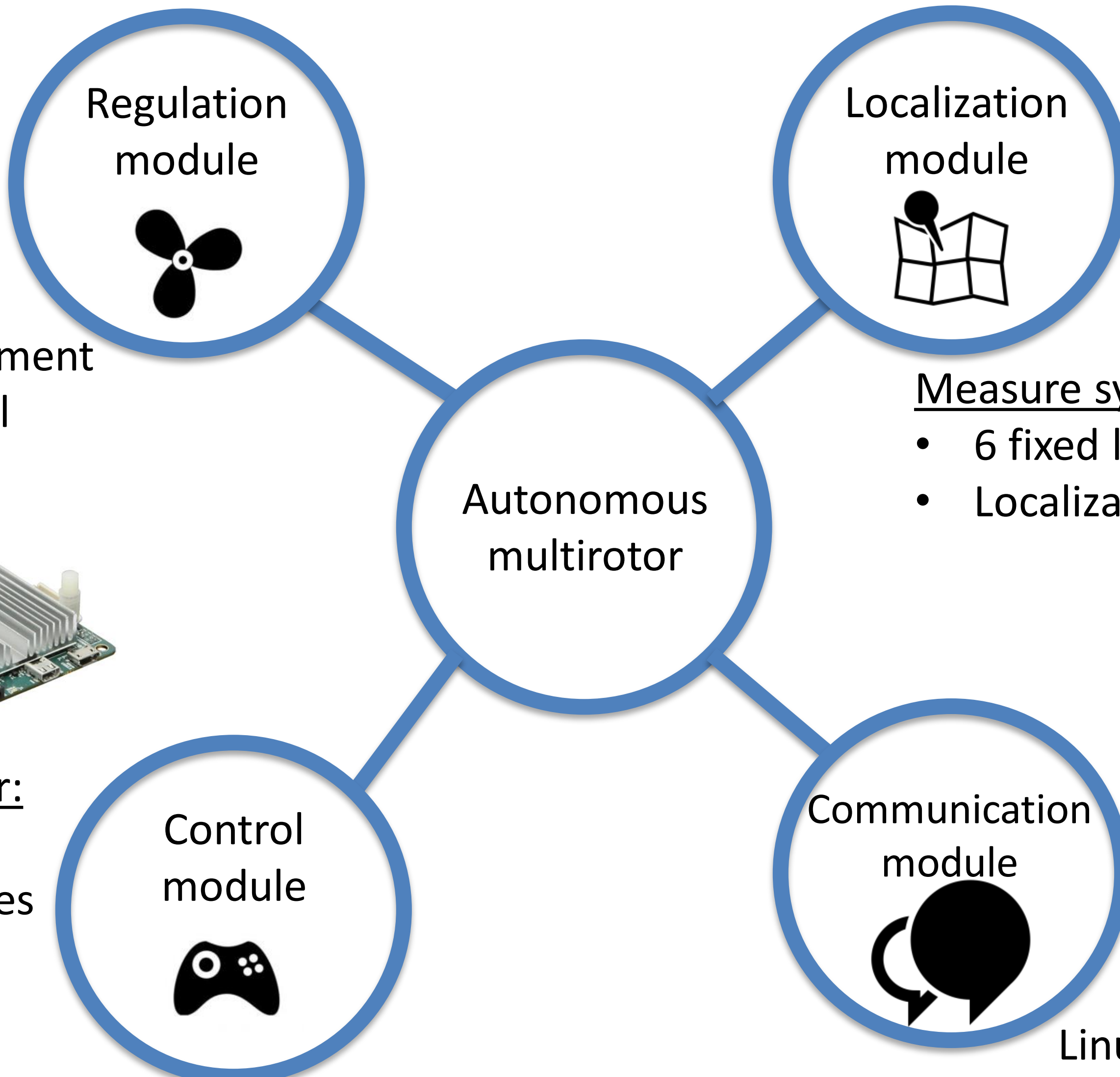
The UAV must be able to autonomously fly in a rectangular room of 10x10 meters with a precision of 5 centimeters (2 inches).

Typical requirements



Our architecture

Flight controller:
- Motor management
- Position control
- Stabilization



Measure system:
- 6 fixed lasers
- Localization algorithm

Onboard computer:
- Brain
- Manage modules
- Tasks list

ROS

Linux software overlay

Regulation module

Flight controller

Measure

IMU = Inertial Measurement Unit, consisting of a magnetometer, an accelerometer, a gyroscope and a barometer

Motor control

- Regulate motor power
- Stabilize the UAV
- Navigate to the setpoint

Communication

- Mavlink
- GPS
- USB
- Serial
- PWM

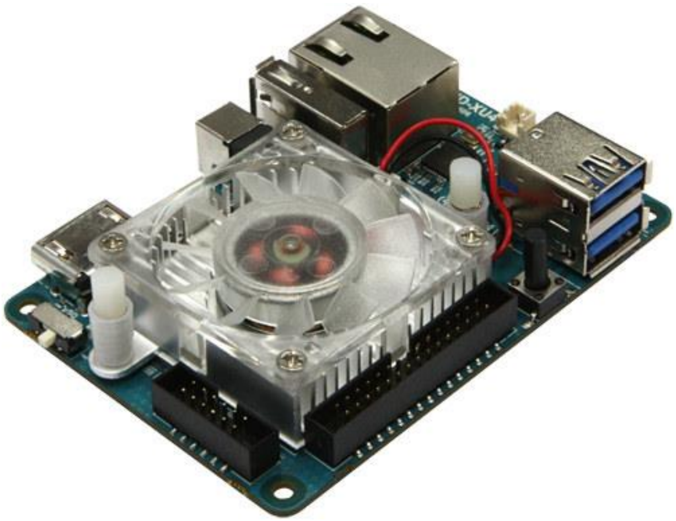
« A flight controller for a multirotor is a printed circuit consisting in a microprocessor, captors and I/O. »

Pixhawk

- 200$
- Open source
- Open hardware
- Can be updated with our regulation loops

Control module

Onboard computer

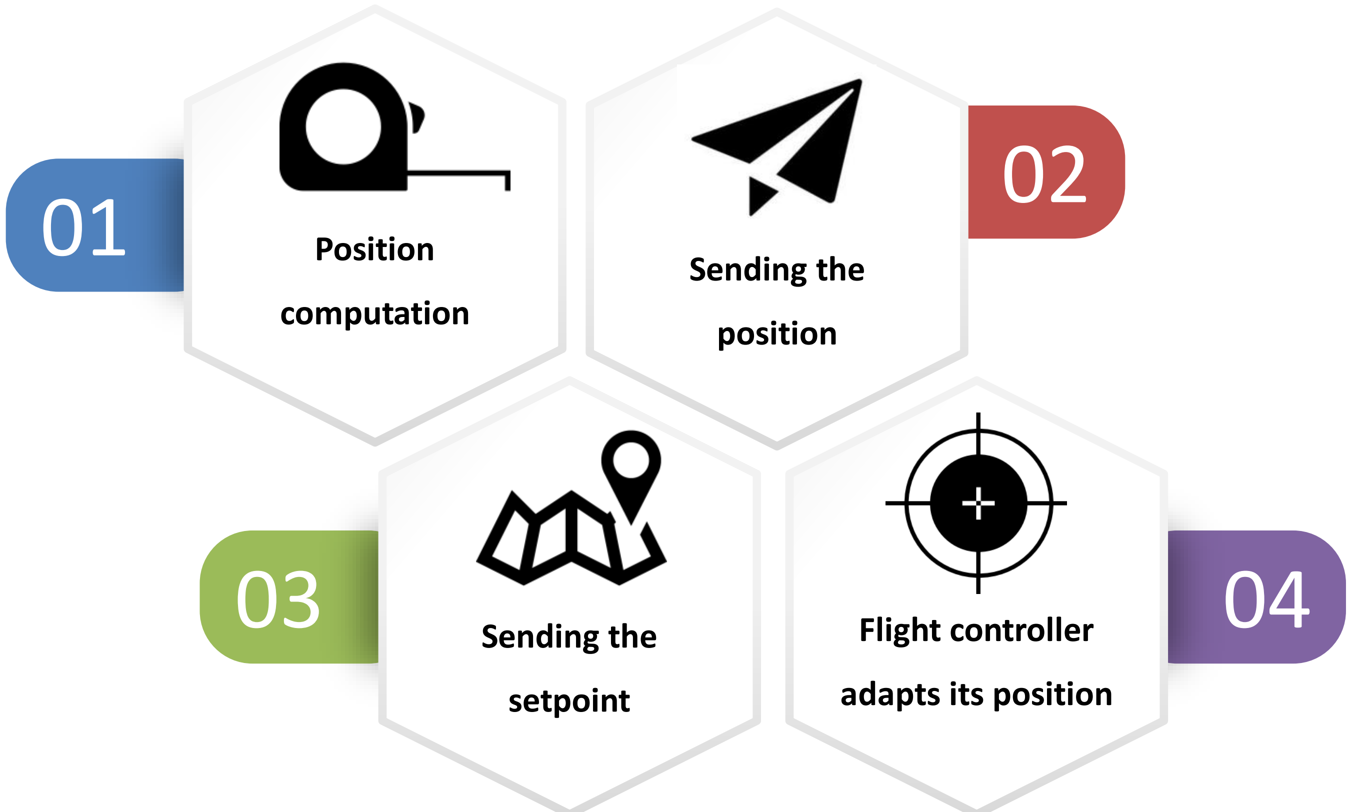


Odroid XU4

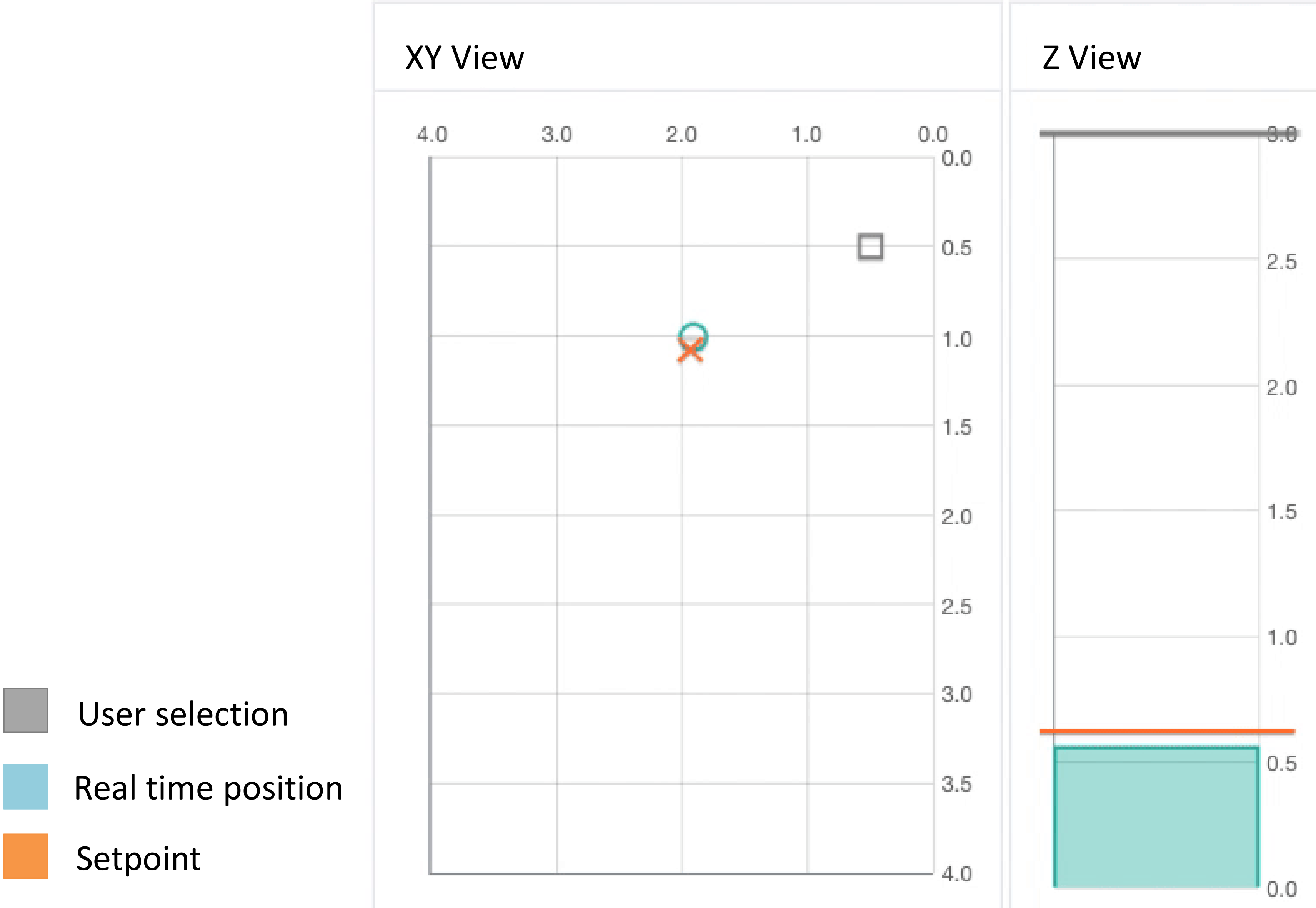
- Operating system: Linux Ubuntu (linaro)
- Open hardware/Open source
- Able to compute CV algorithms
- Possible graphic interface
- 2GB RAM
- Exynos 5422, octocore (LITTLE.big)
- Price: 79$ (without TVA nor shipping)

- Used as an onboard computer
- The differences between models are mainly about performances

Control process



Real time user-friendly interface



Communication module

ROS

> *ROS is a software overlay under Linux. The purpose of ROS is to provide common robotic libraries and a simplification of the inter process communication and hardware abstraction layers.*

Advantages:

- Specialized in robotics
- Lots of libraries
- Provide modularity to any project
- Simplify any type of communications
- Community driven
- Open-source
- Can be used even for commercial use for most of components
- Plug'n'play

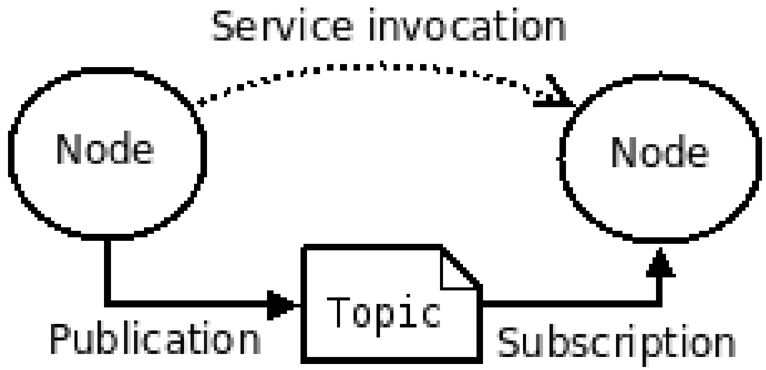
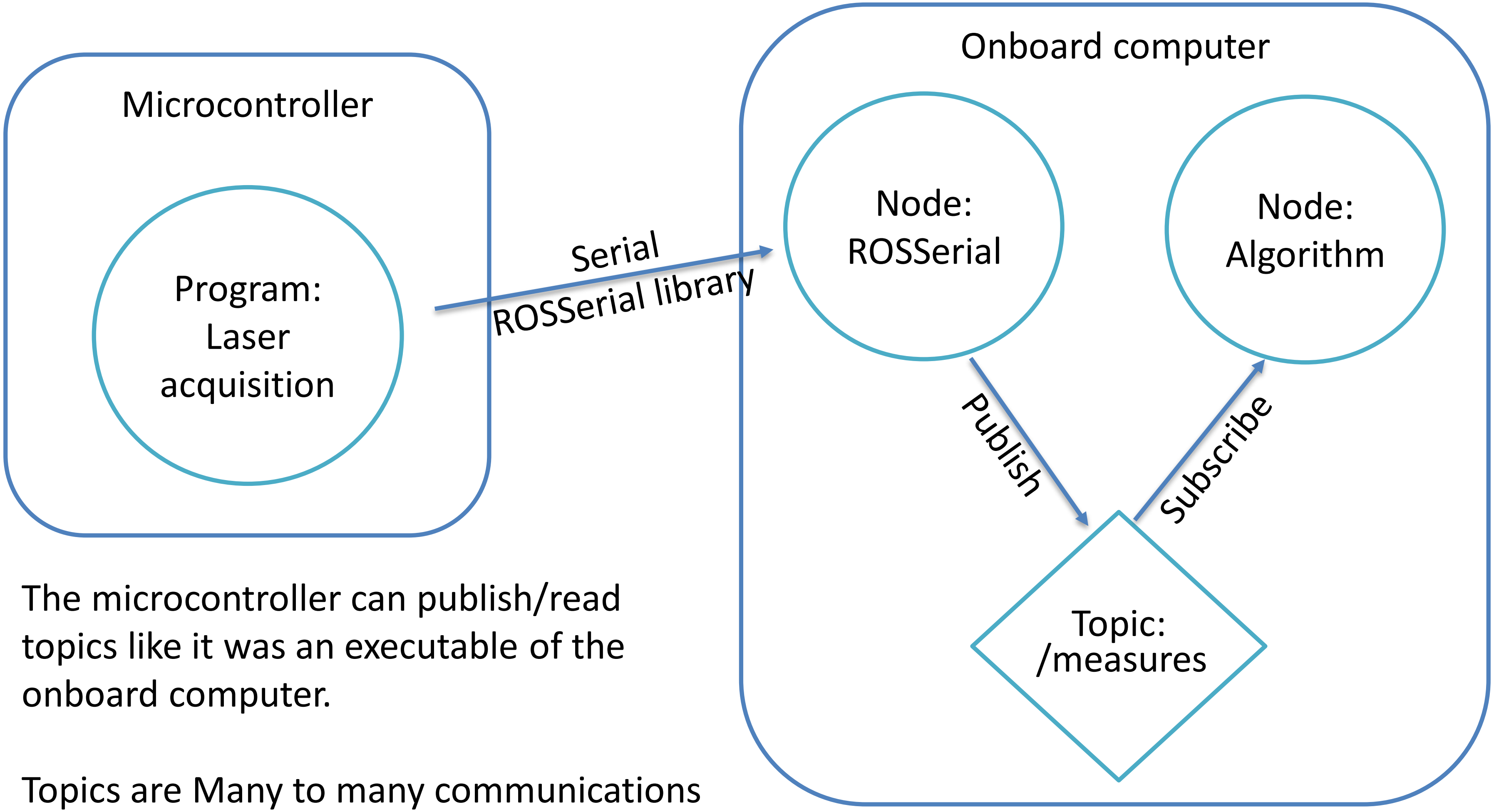


Figure : ROS communication philosophy

Example of ROS: Laser acquisition



The microcontroller can publish/read topics like it was an executable of the onboard computer.

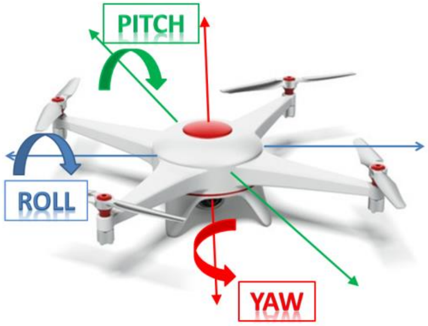
Topics are Many to many communications

Localization module

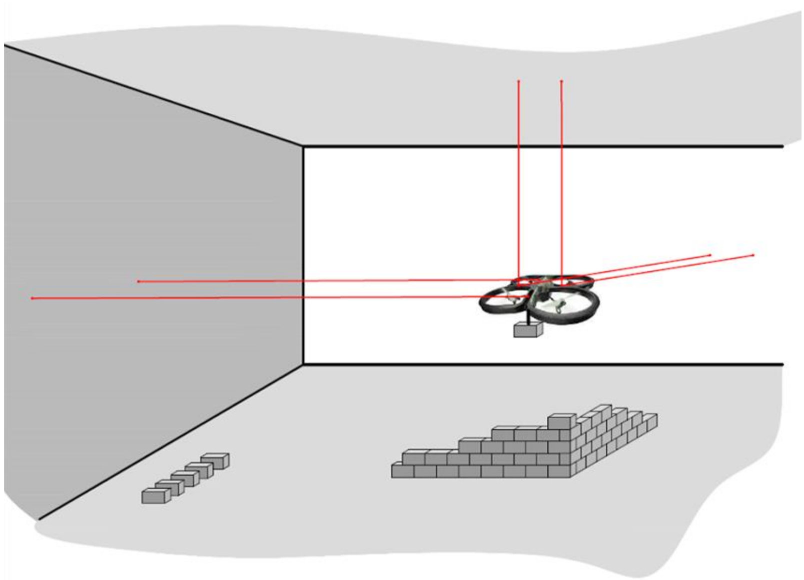
Fixed laser method

6 lasers pointing 3 reference walls

Actual prototype



- Yaw dependant
- Risk to loss the reference walls
- No room evolution
- No obstacle avoidance
- Faster to develop



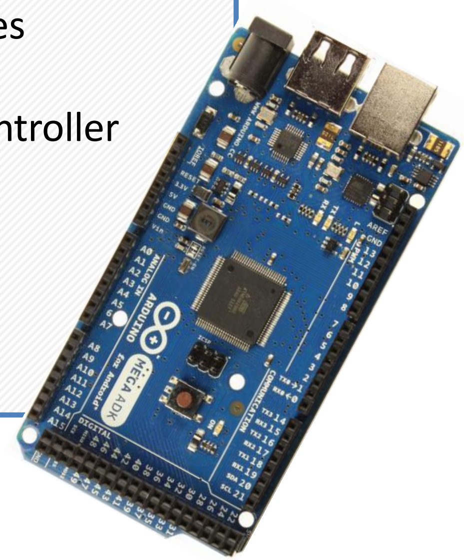
Lasers

Lidar Lite v2:

- Precision: 2.5 cm, 1 inch
- Maximal acquisition frequency: 500 Hz
- Weight: 22 gr
- Range: 1- 40 m, 44 yards
- Class 1: Not hazardous for eyes

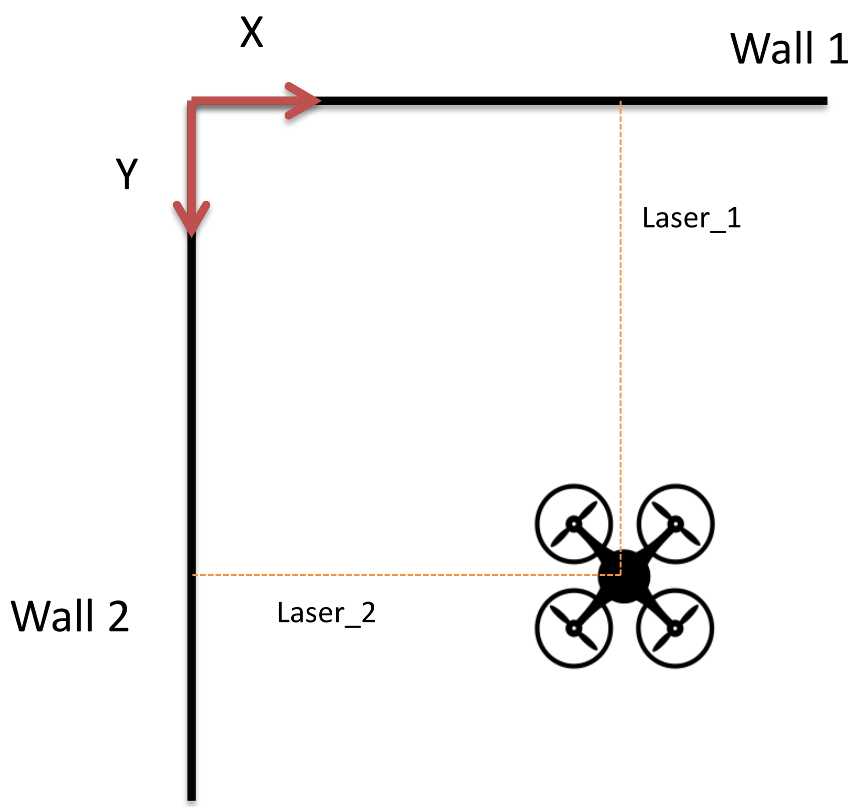
- Acquisition though a microcontroller
- Protocols: **I2C**, PWM

- Price: 100$



Localization algorithm

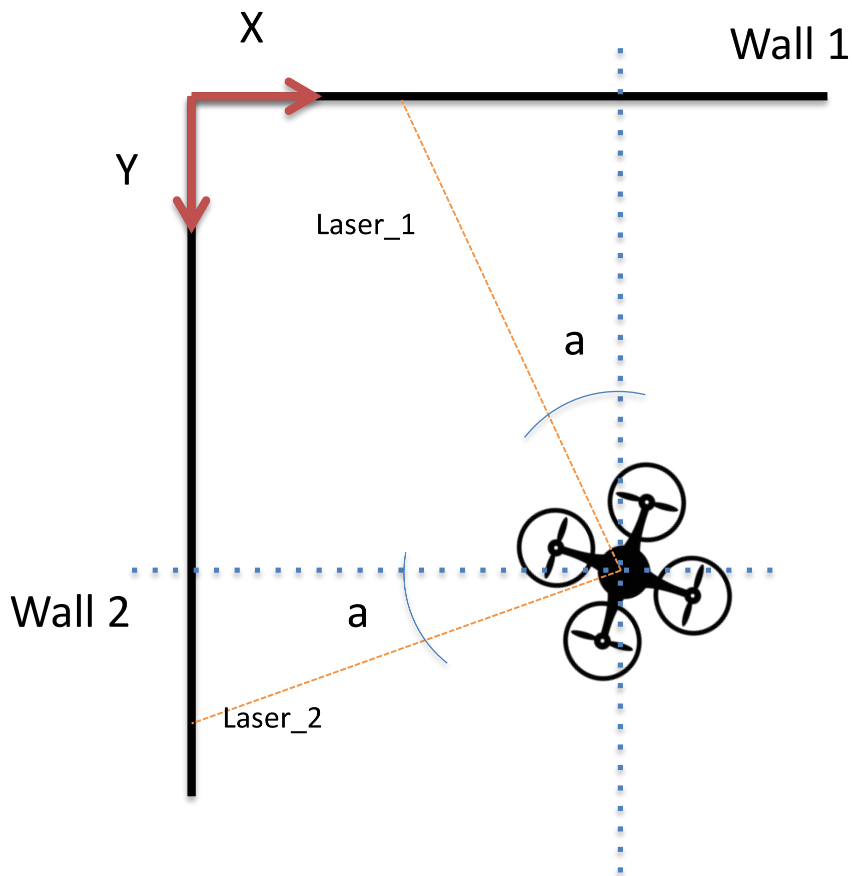
2D



X position = Laser 2 measure
Y position = Laser 1 measure

Localization algorithm

2D

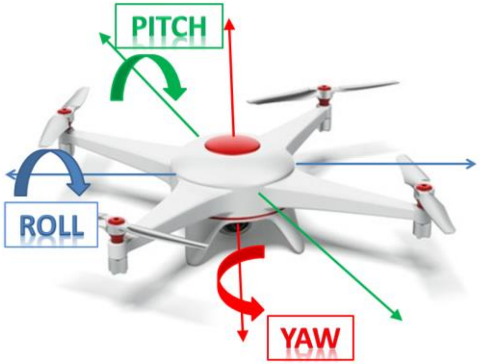


X position = Laser 2 measure * **cos(a)**
Y position = Laser 1 measure * **cos(a)**

Localization algorithm

3D

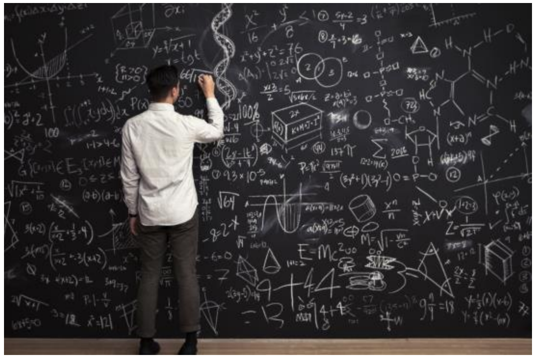
- Same logic
- More complex equations
- 100Hz to filter afterwards



6 measures

UAV orientation

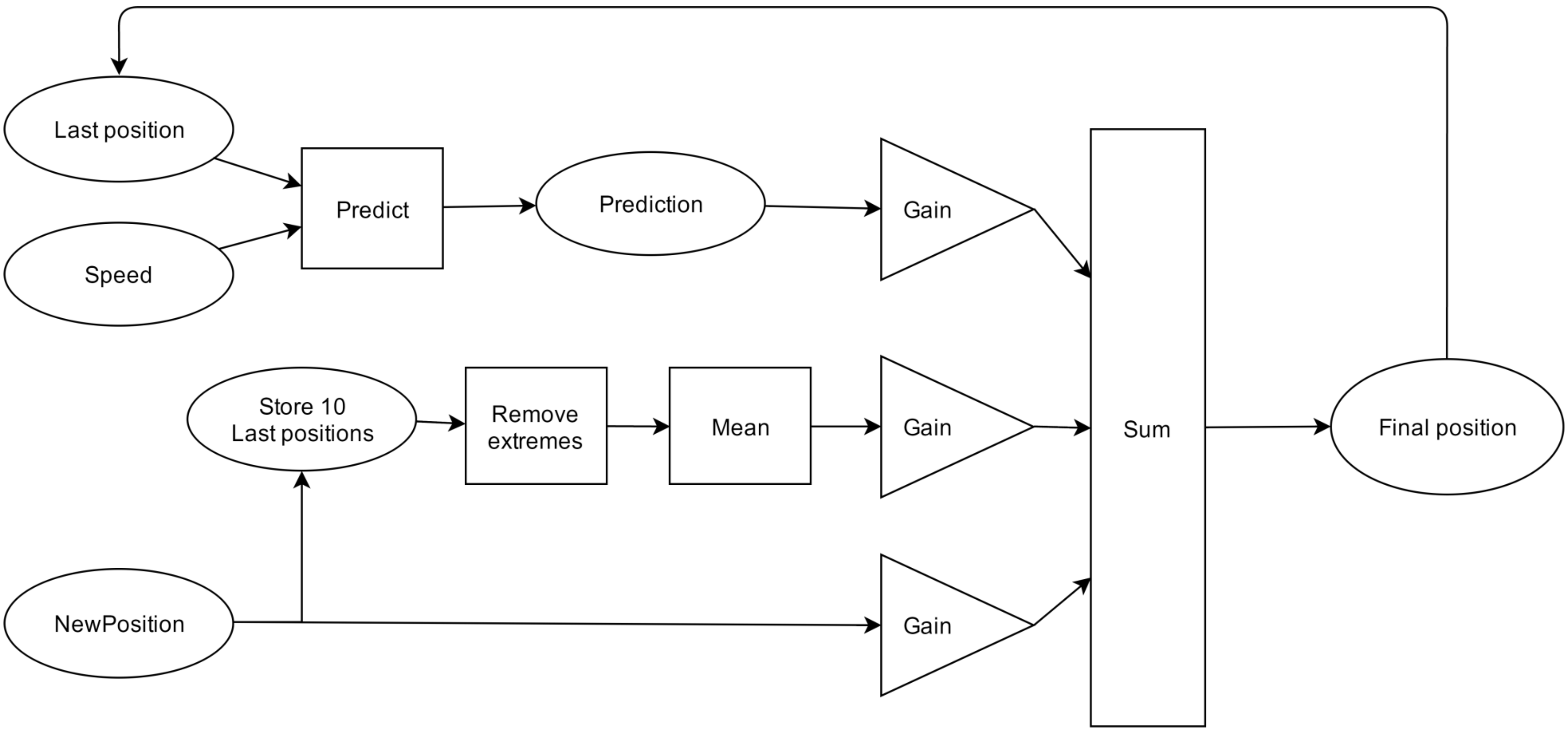
Laser position
on the UAV



Equations

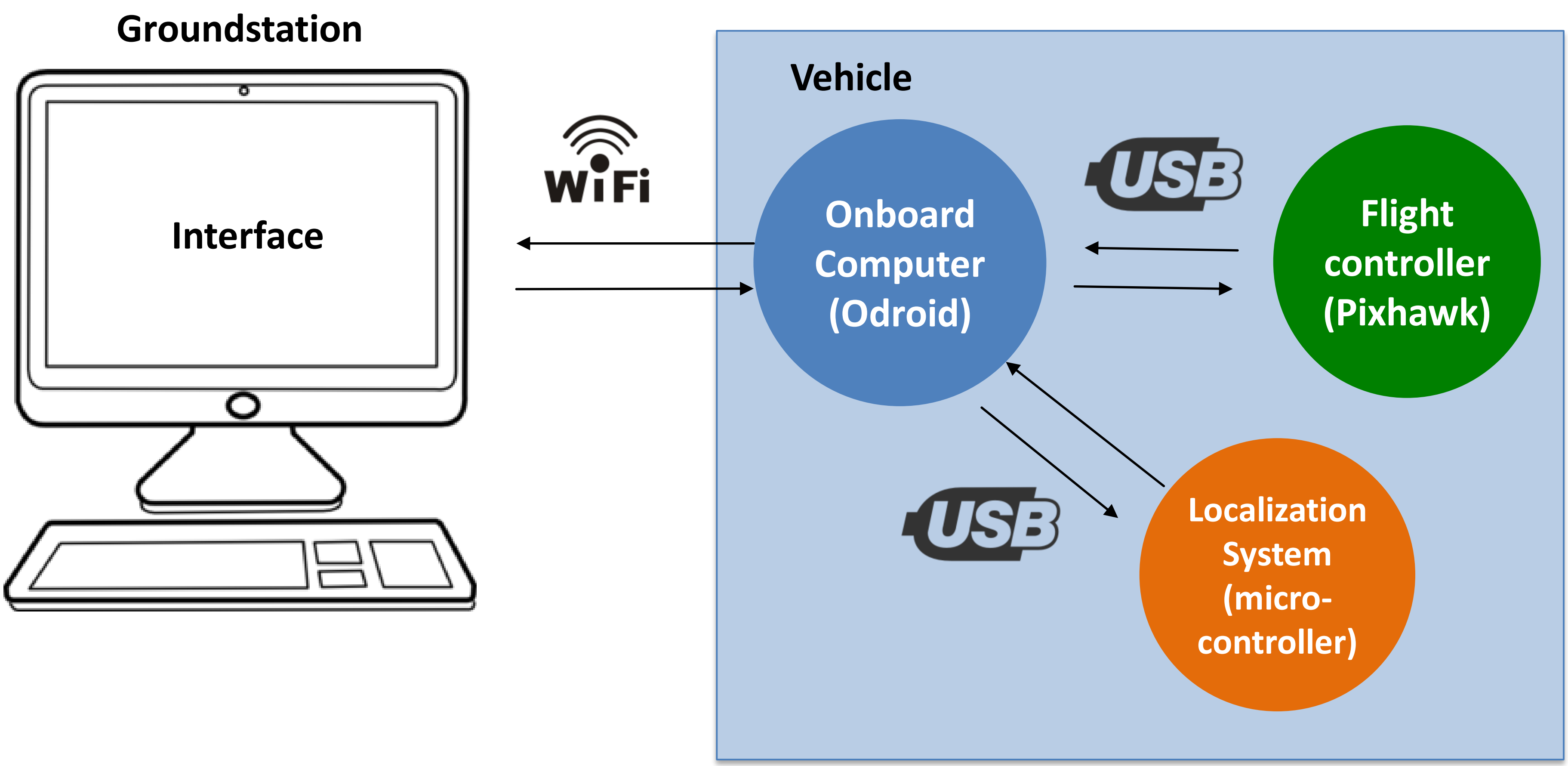
- XYZ position
- Yaw angle

Localization output filter



Changing gains, we can choose between an « instant position », using the prediction and the feedforward or the « mean position » to achieve a better positioning when we have a low dynamic.
There are additional low pass filters on laser inputs (one pole IIR filter)

Global architecture



Performances

Static, algorithm

Angular precision:

Pitch & Roll: $\sigma = 0.05°$

Yaw: $\sigma = 0.5°$

Linear precision:

$\sigma = 0.82$cm (0.3 in)

Dynamic, flying

Angular precision:

Yaw: $\sigma = 2.1°$

Linear precision:

$\sigma = 2.8$cm (1.1 in)

PID needs better tuning

PID: Proportional, Integral, Derivative control loop, with optional Feedforward gain.
This represents about 20 parameters because of the 6 degrees of liberty of the UAV

Final results of the system

Pros

- Ready to use for next year
- Pretty fast to install
- 2 inches precision with fine PID tuning
- Modular architecture
- Development workspace
- Friendly user interface

Cons

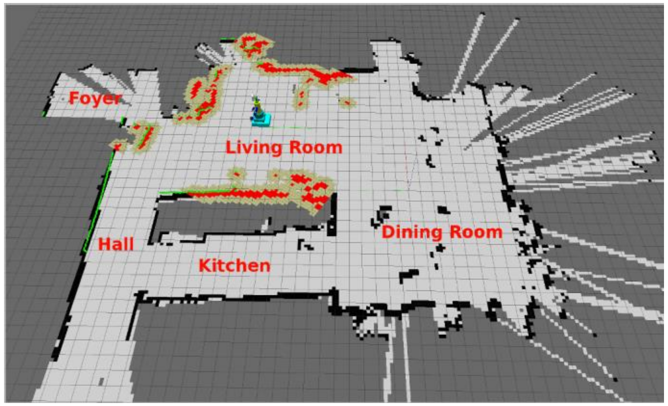
- Rectangular room
- No environment evolution
- Not enough robust for production
- No security (third party with malicious intent)
- No PID tuning
- No obstacle avoidance
- Overweight of the prototype

Perspectives

1. Increase the prototype size

2. PID tuning

3. Sensor fusion
   a) Delay compensation
   b) RTK
   c) Camera



4. Mapping of the environnement