BOARR

https://github.com/Gipsa-lab-PFP/BOARR

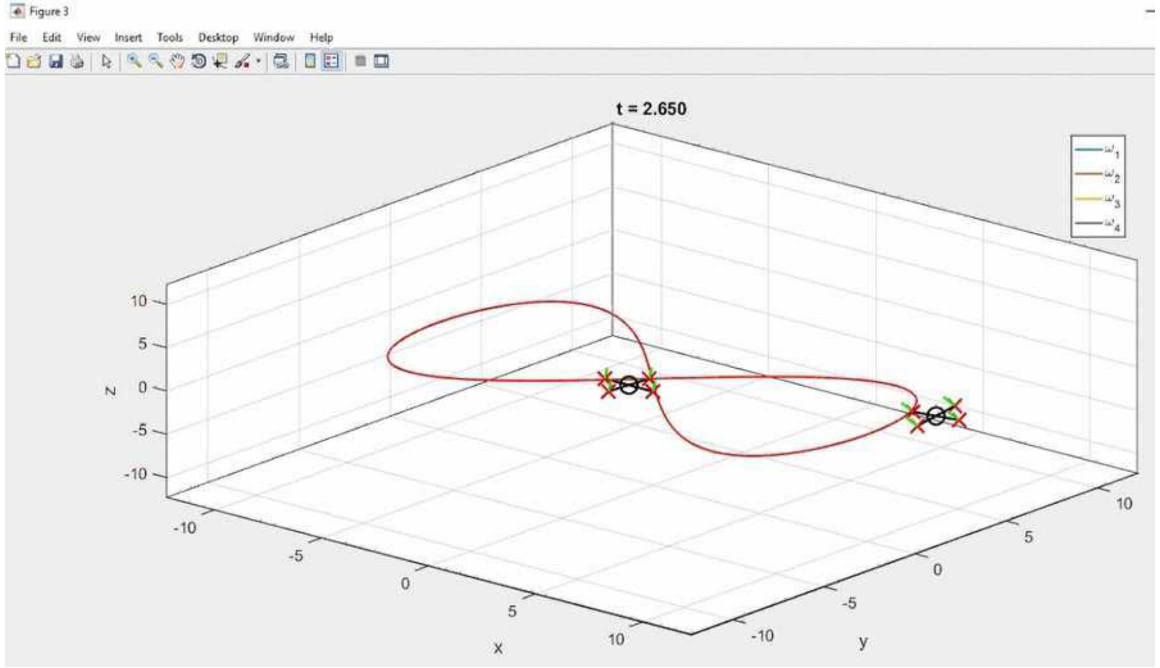
A Benchmark for quadrotors Obstacle Avoidance using ROS and RotorS

T. TEZENAS DU MONTCEL, A. NEGRE, E. GOMAZ-BALDERAS, N. MARCHAND

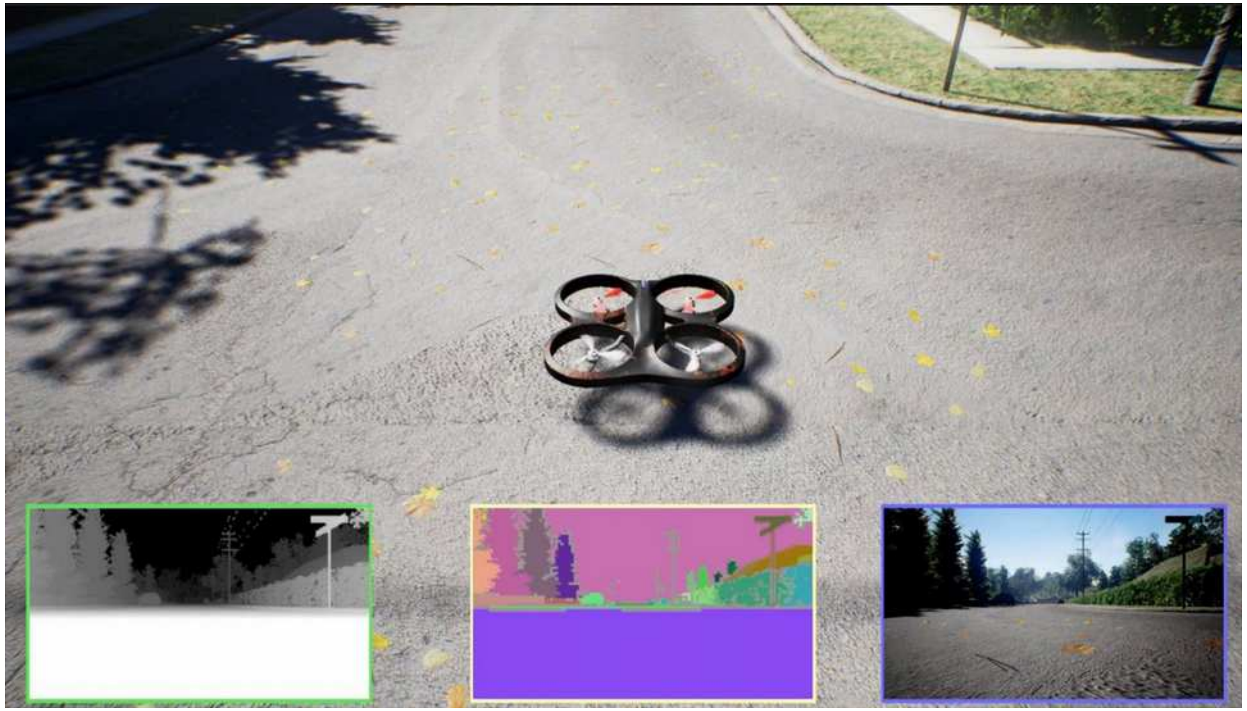
Quadrotor obstacle avoidance

Simulator choice

- MATLAB + Simulink

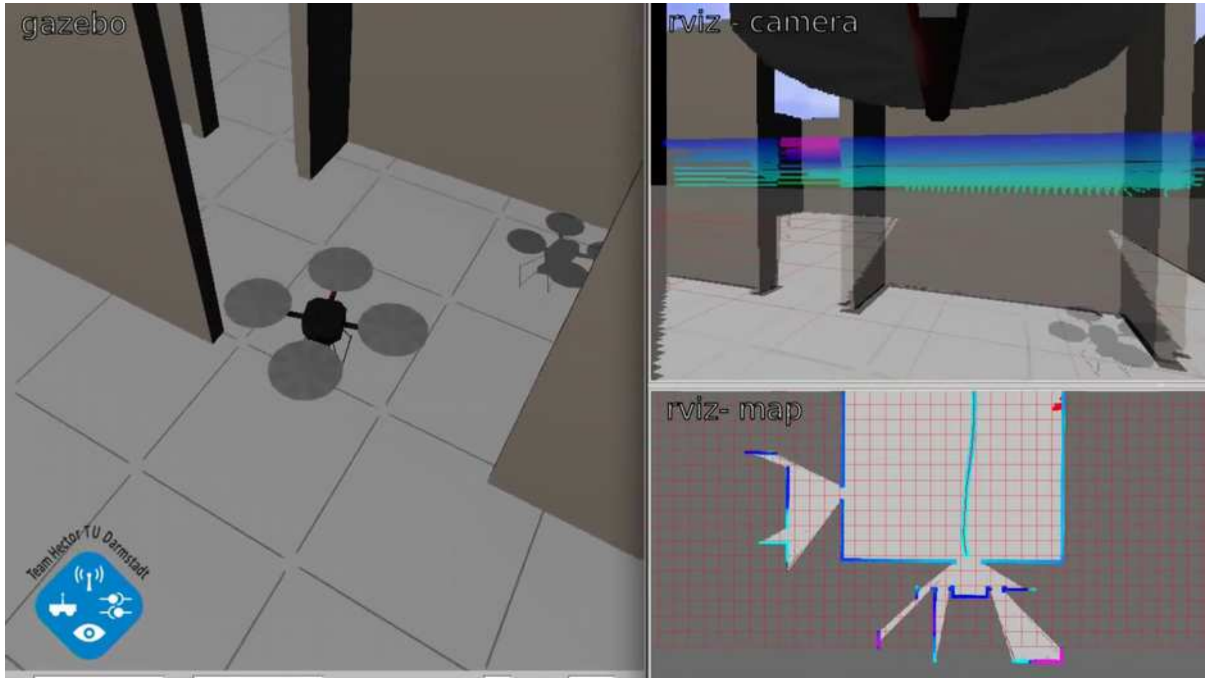


- Microsoft AirSim (2017)

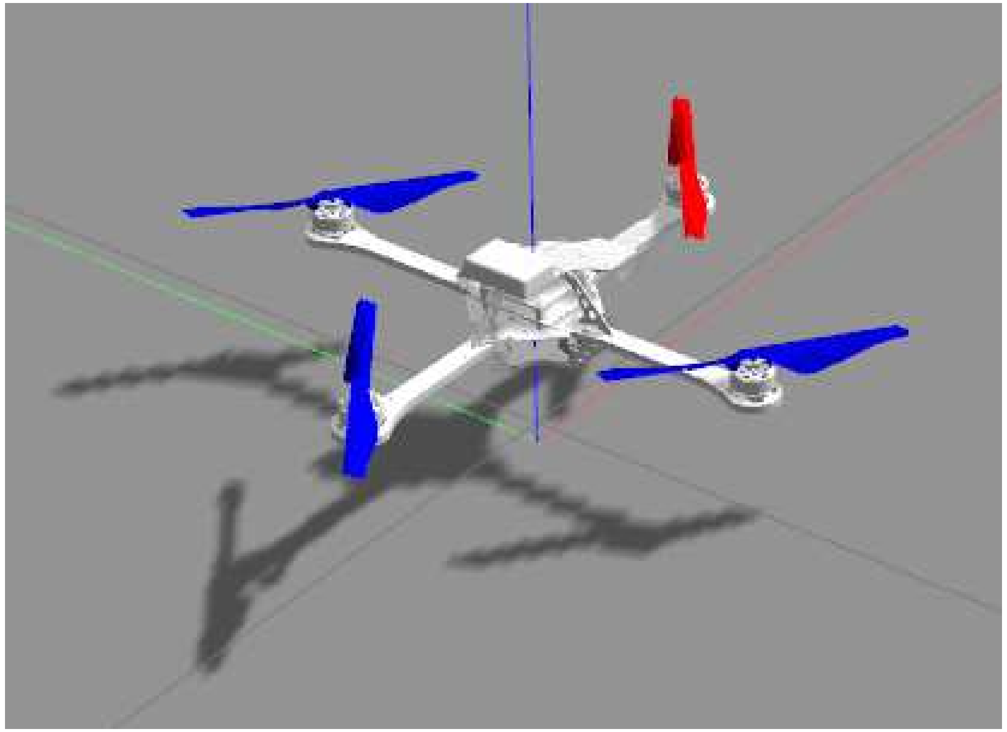


Simulator choice

- Hector Quadrotor (2012)

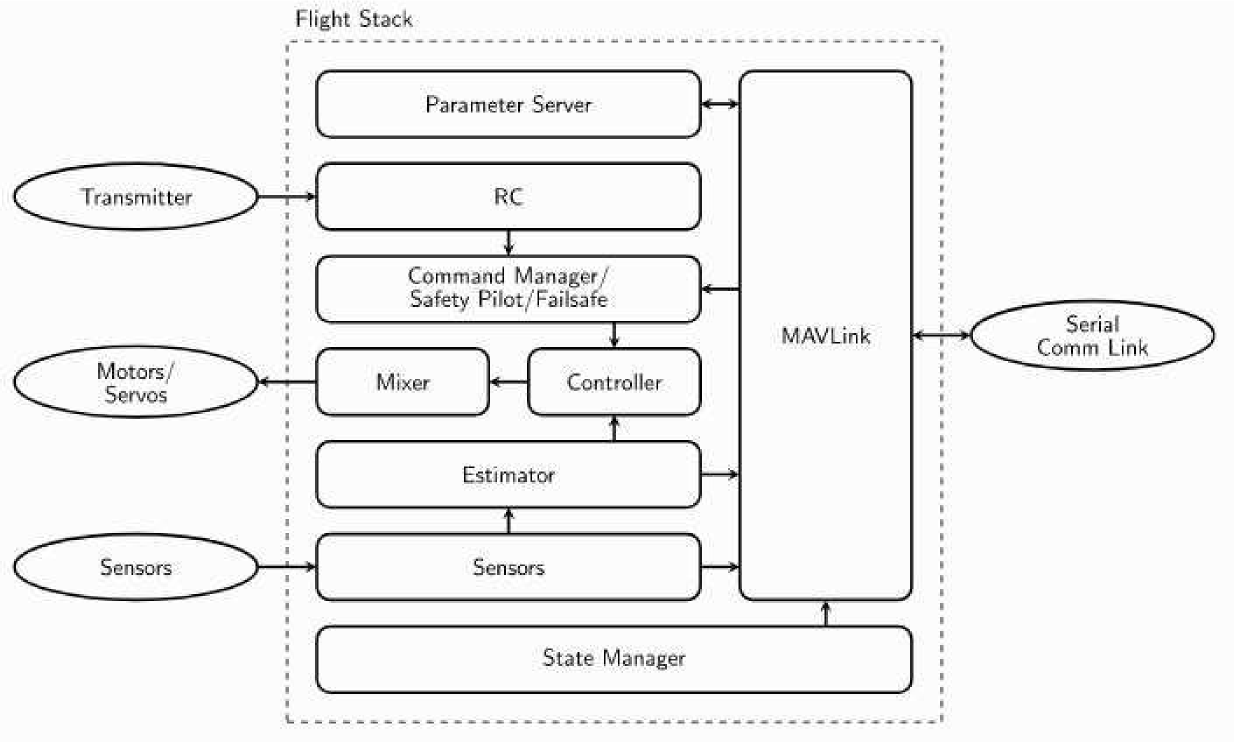


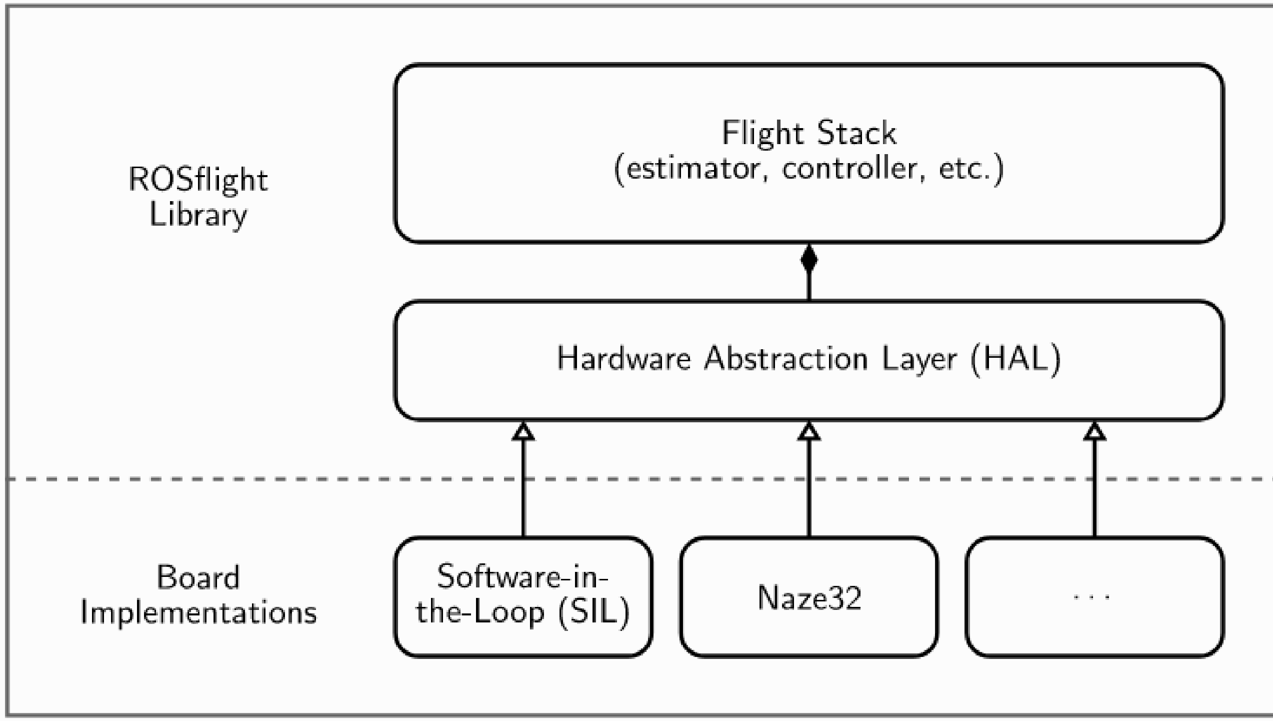
- RotorS (2016)



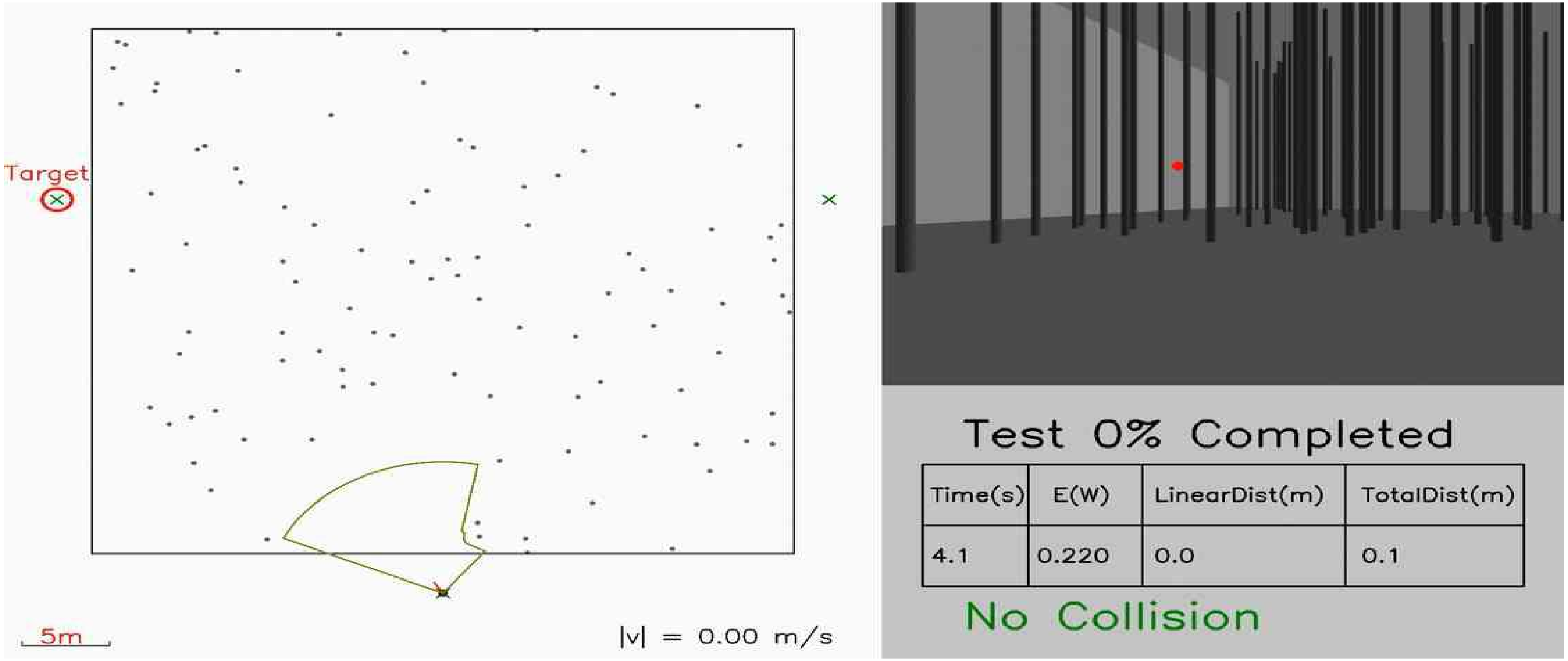
Simulator choice

- ROSflight (2018)





Algorithm Evaluation



Test 0% Completed

Time(s)	E(W)	LinearDist(m)	TotalDist(m)
4.1	0.220	0.0	0.1

No Collision

Algorithm Evaluation

A simulated benchmark ?

The BOARR benchmark

Step-by-step guide

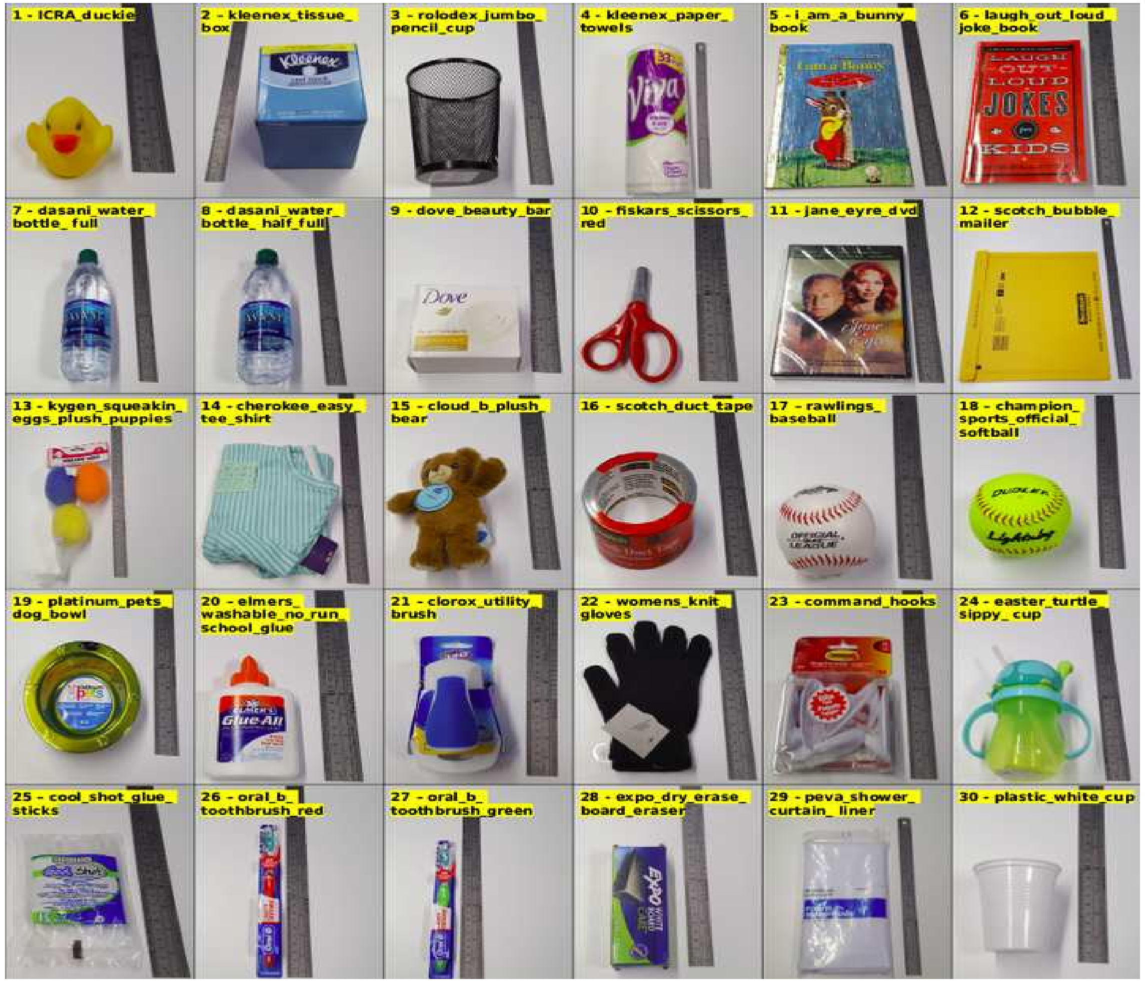
Conclusions

Objectives

- Quantitative evaluation of each algorithm
- Comparison of algorithm proposals
- Clarify the state of the art
- A step toward compatibility
- A step toward reproducibility

Other comparison tools

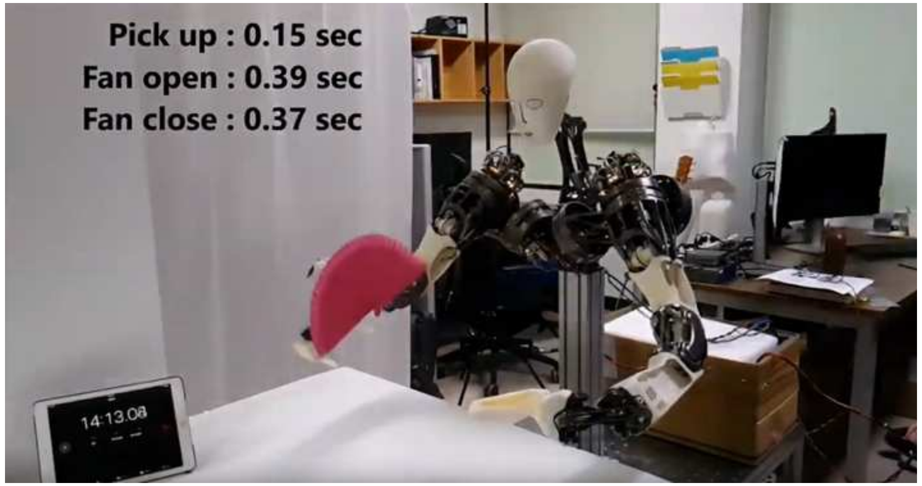
- Real life benchmarks :
  - Set strict protocols
  - Not always possible


1 - ICRA_duckie
2 - kleenex_tissue_box
3 - rolodex_jumbo_pencil_cup
4 - kleenex_paper_towels
5 - i_am_a_bunny_book
6 - laugh_out_loud_joke_book
7 - dasani_water_bottle_full
8 - dasani_water_bottle_half_full
9 - dove_beauty_bar
10 - fiskars_scissors_red
11 - jane_eyre_dvd
12 - scotch_bubble_mailer
13 - kygen_squeakin_eggs_plush_puppies
14 - cherokee_easy_tee_shirt
15 - cloud_b_plush_bear
16 - scotch_duct_tape
17 - rawlings_baseball
18 - champion_sports_official_softball
19 - platinum_pets_dog_bowl
20 - elmers_washable_no_run_school_glue
21 - clorox_utility_brush
22 - womens_knit_gloves
23 - command_hooks
24 - easter_turtle_sippy_cup
25 - cool_shot_glue_sticks
26 - oral_b_toothbrush_red
27 - oral_b_toothbrush_green
28 - expo_dry_erase_board_eraser
29 - peva_shower_curtain_liner
30 - plastic_white_cup

ACRV picking benchmark

Other comparison tools

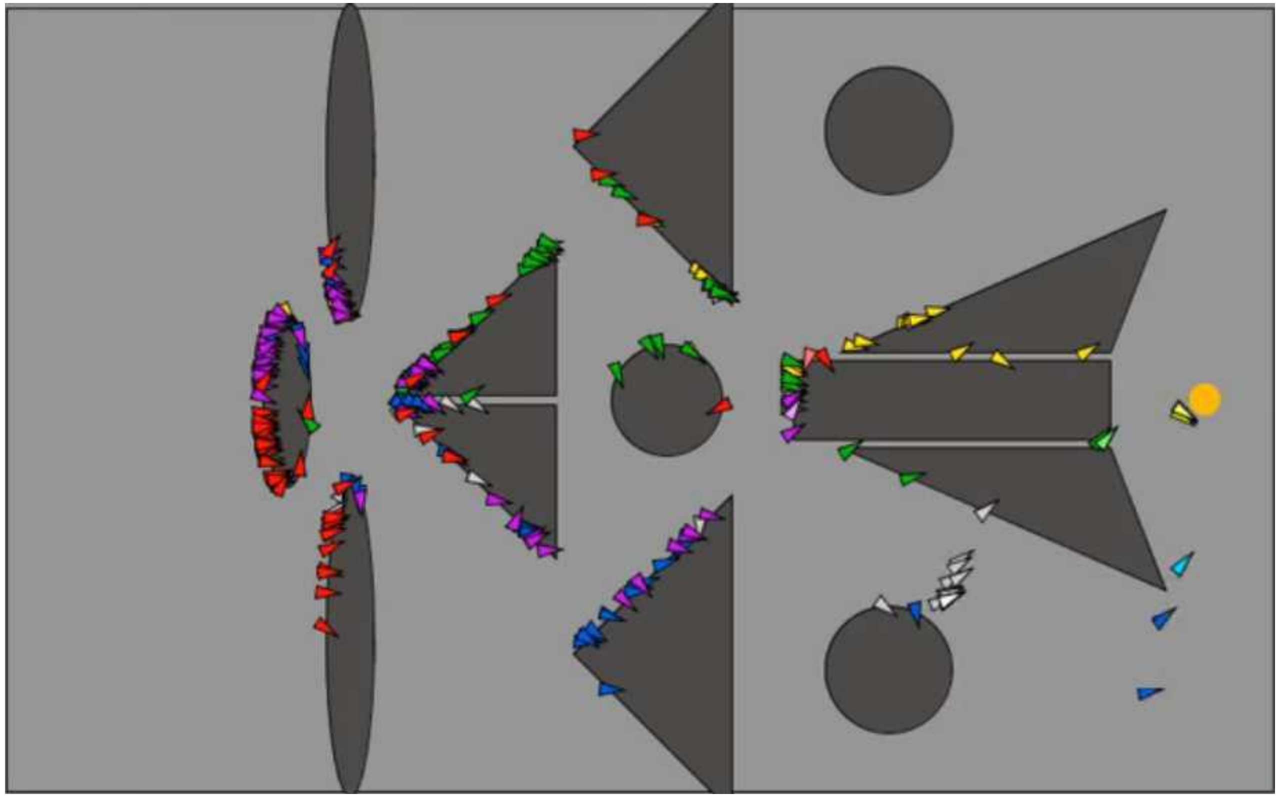
- Challenges :
  - DARPA Grand Challenge
  - Yearly competitions at ICRA, IROS…
  - Autonomous Drone Racing Competition, IROS 2016-2018



Advantages of simulation

- Allow statistical Analysis

- Allow tests during early phases of a project



A Gazebo-ROS benchmark ?

The BOARR Benchmark

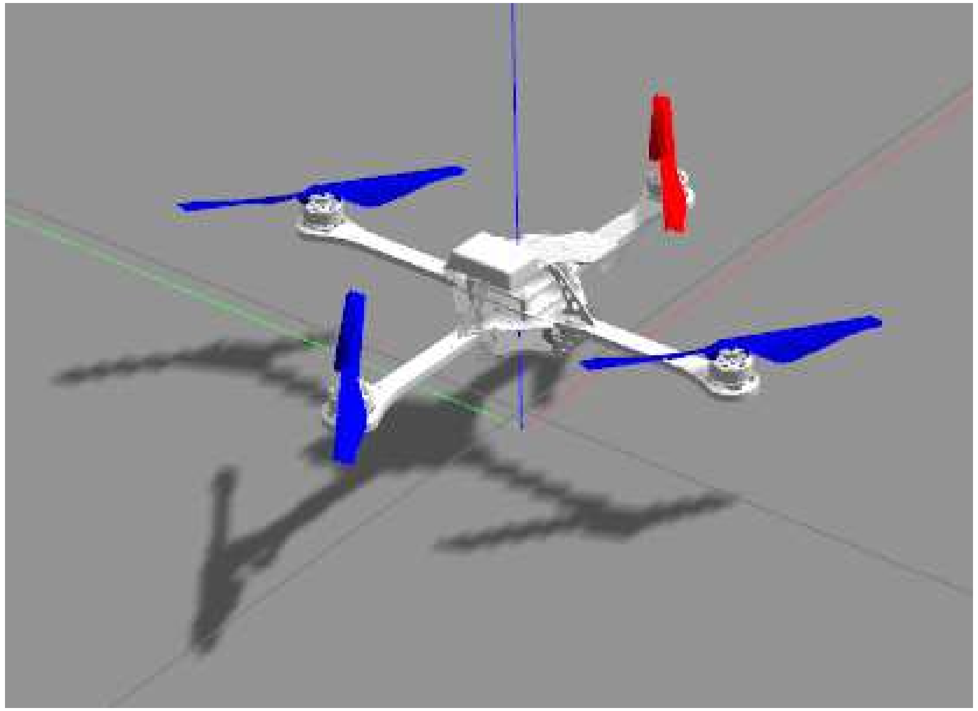
Step-by-step guide

Conclusions

Using RotorS for genericity

- Most common quadrotor simulator
- Most common quadrotor size and weight
- Multiple control options are proposed

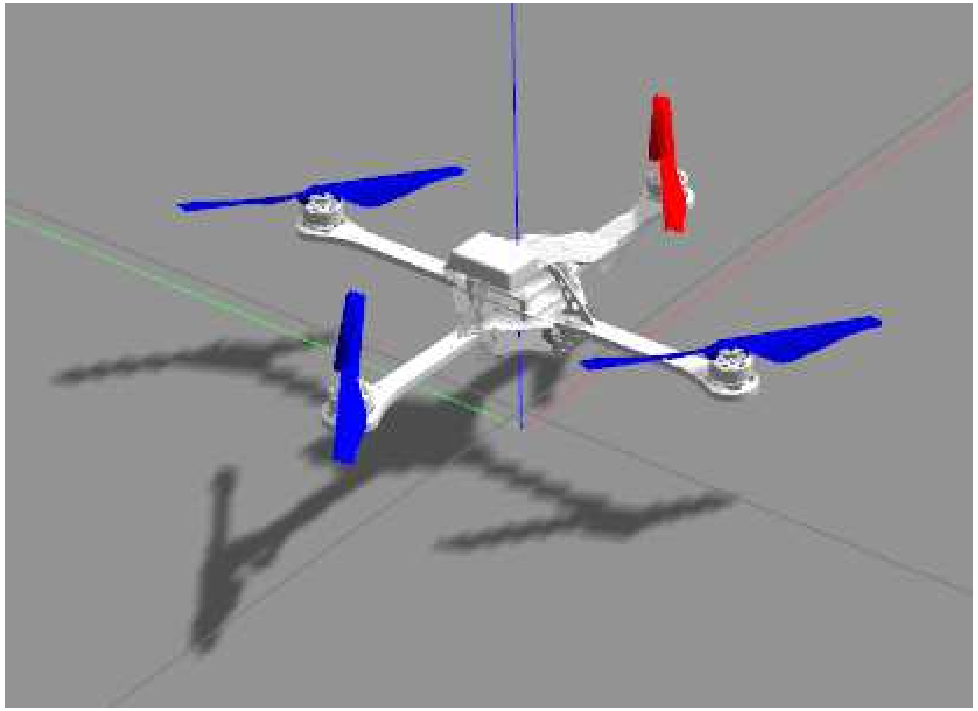
  from position control to motor control
- Multiple Cameras and Depth sensors



Using ROS for genericity

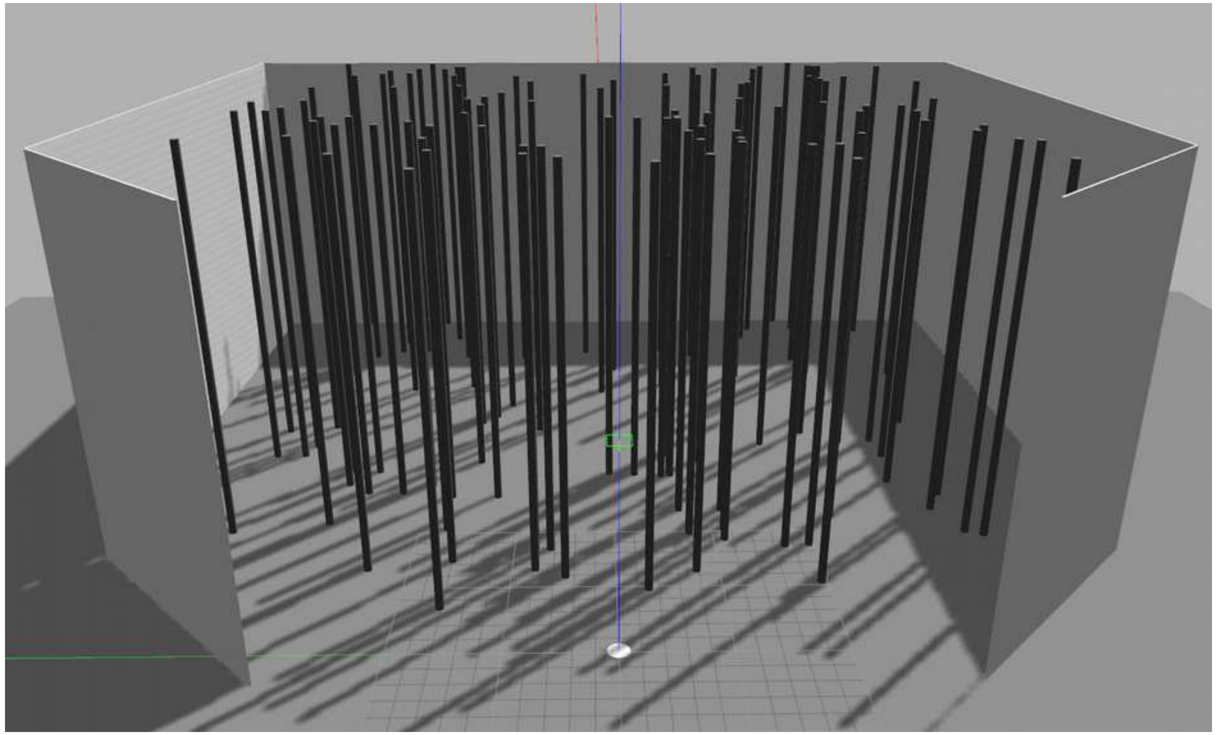
Standard ROS messages for sensors (inputs) :

- sensor_msgs/Imu
- sensor_msgs/PointCloud2
- sensor_msgs/Images
- sensor_msgs/NavSatFix
- sensor_msgs/MagneticField



Generated Worlds

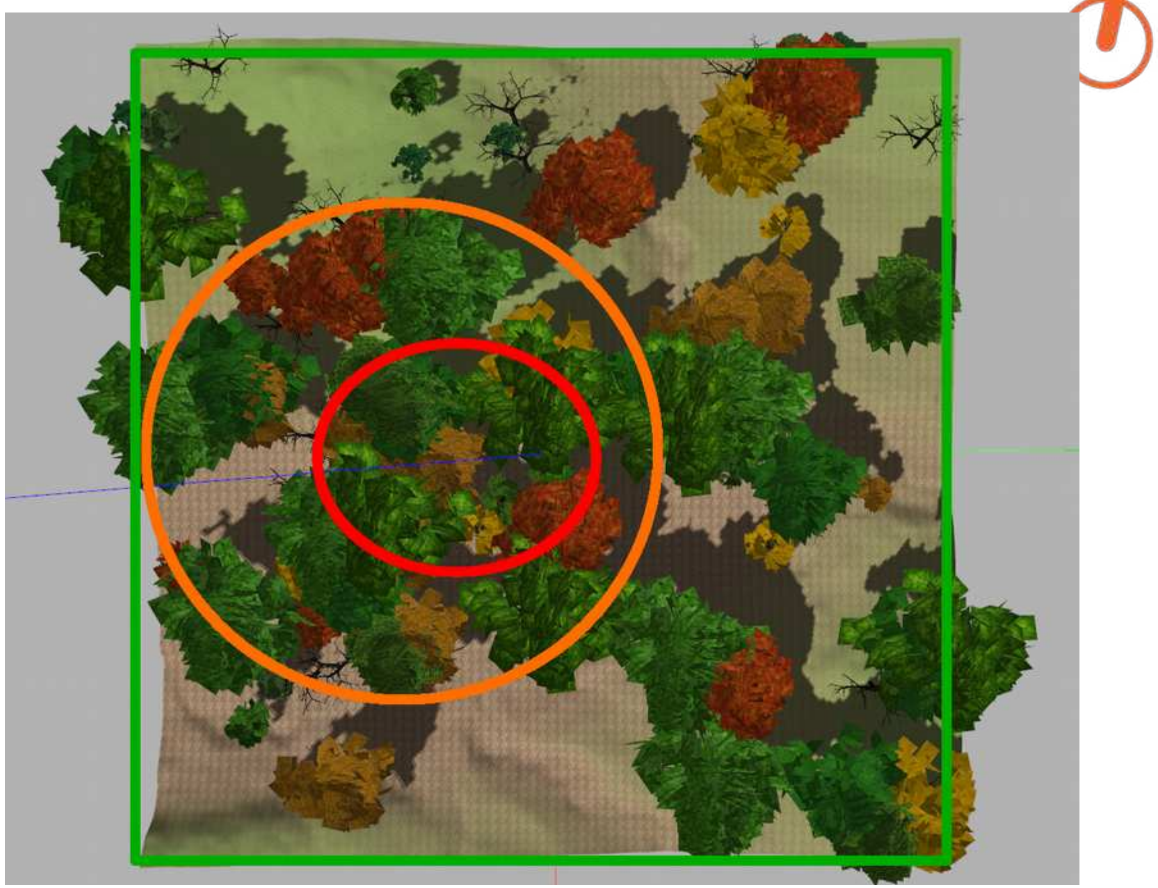
- Geometric :
  - Unrealistic, perfect sensing
  - lightweight



Generated Worlds

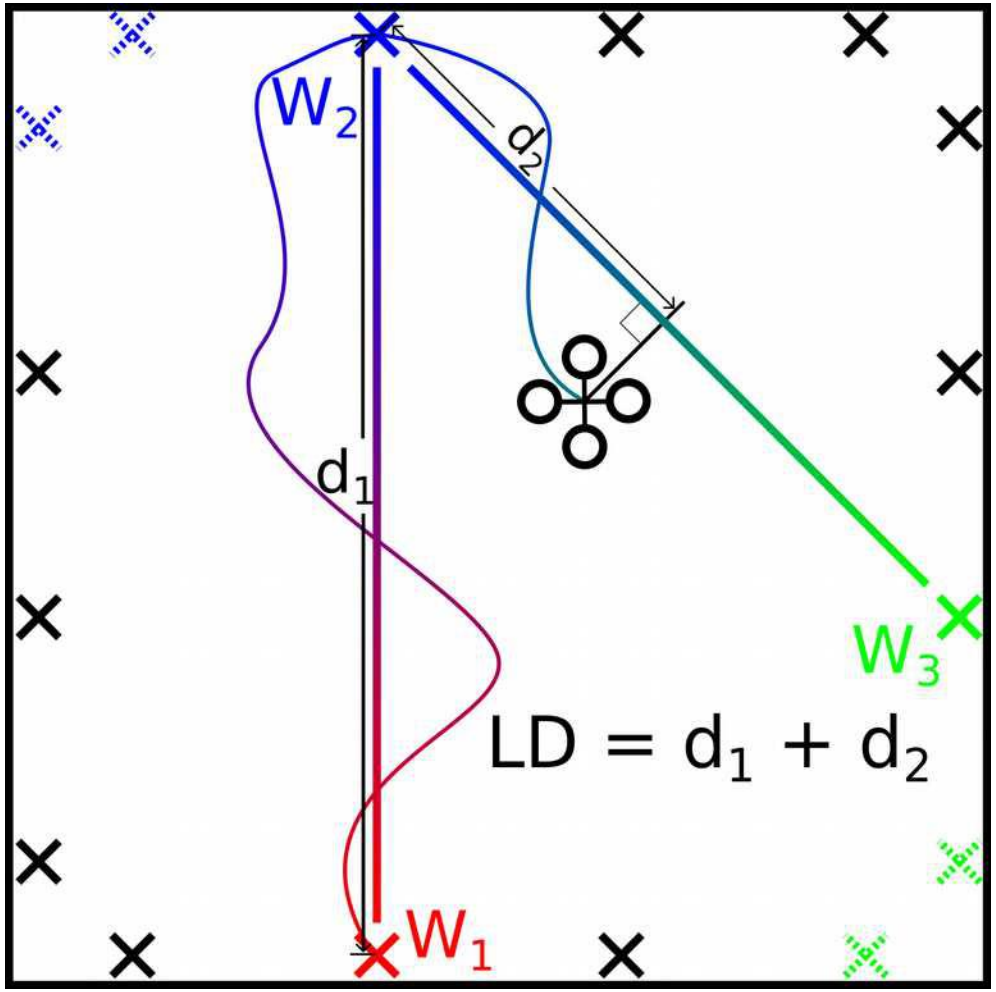
Forests :

- Multi-density forests
- As light as possible : low-poly trees, same trees rotated
- Automatic generation of multiple parametrized worlds



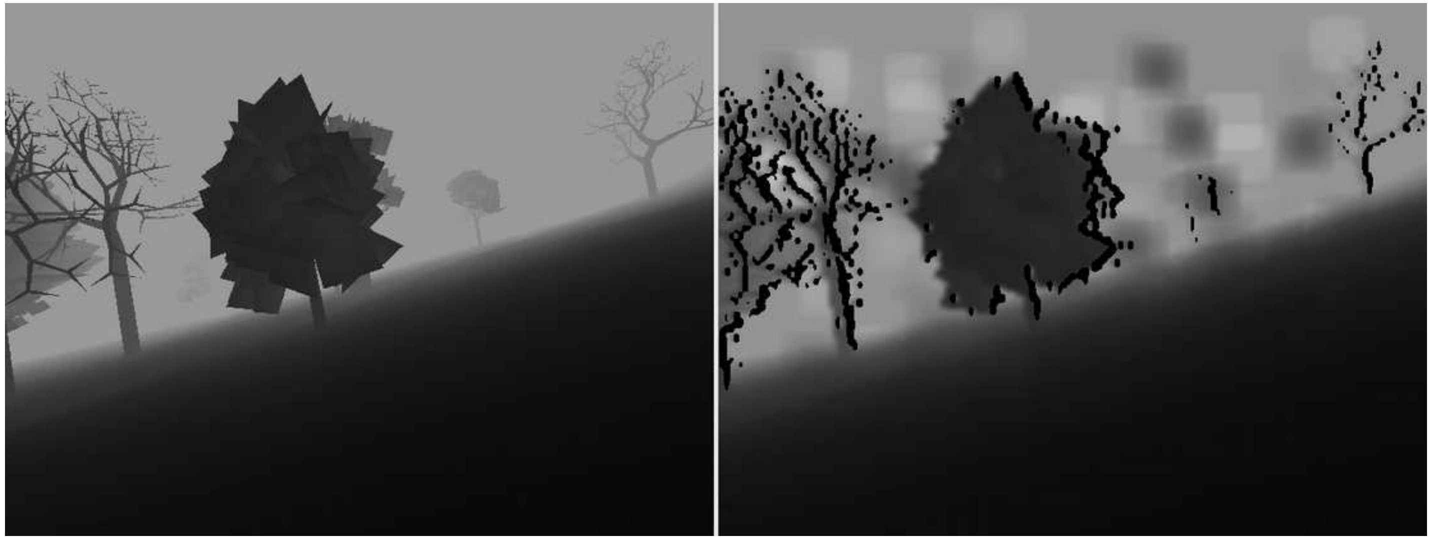
Unit Test

Single Goal : Reach 1km of flight while avoiding collision and reaching a set of predefined way-points



Increase the difficulty by adding disturbances

- Adding realistic disturbances is different from having a realistic render
  - Noise nature and level comparable to natural noise
  - Simulated wind impact comparable to real wind



Statistical Analysis

- Non deterministic environment :
  - ROS-Gazebo by itself is non deterministic
  - Simulated wind and sensors noises are non deterministic

- Comparable tests :
  - Wind and sensor noises have the same profiles across multiple tests

Main Indicator

- Probability to fly 1km without a collision

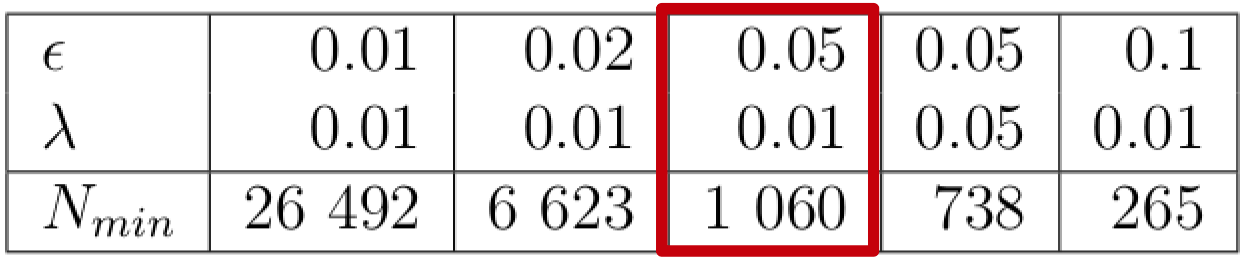
$$\hat{p} = \frac{1}{N}\sum_{i=1}^{N} X_i \quad \text{with } X_i = \begin{cases} 1, & \text{if No Collision on Test } i \\ 0, & \text{otherwise} \end{cases}$$

  - Probabilistic bound (Chernoff bound)

$$Pr(|p - \hat{p}| \leq \epsilon) \geq 1 - \lambda$$

$$N > \frac{\ln\left(\frac{2}{\lambda}\right)}{2\epsilon^2}$$

  - The bound precision depends on the number of tests



ϵ	0.01	0.02	0.05	0.05	0.1
λ	0.01	0.01	0.01	0.05	0.01
N_{min}	26 492	6 623	1 060	738	265

Secondary indicators

- All successful tests :
  - Average speed
  - Average time
  - Average traveled distance
  - Average energy consumed
- On failed tests :
  - Average Linear Distance

A Gazebo-ROS benchmark ?

The BOARR Benchmark

Step-by-step guide

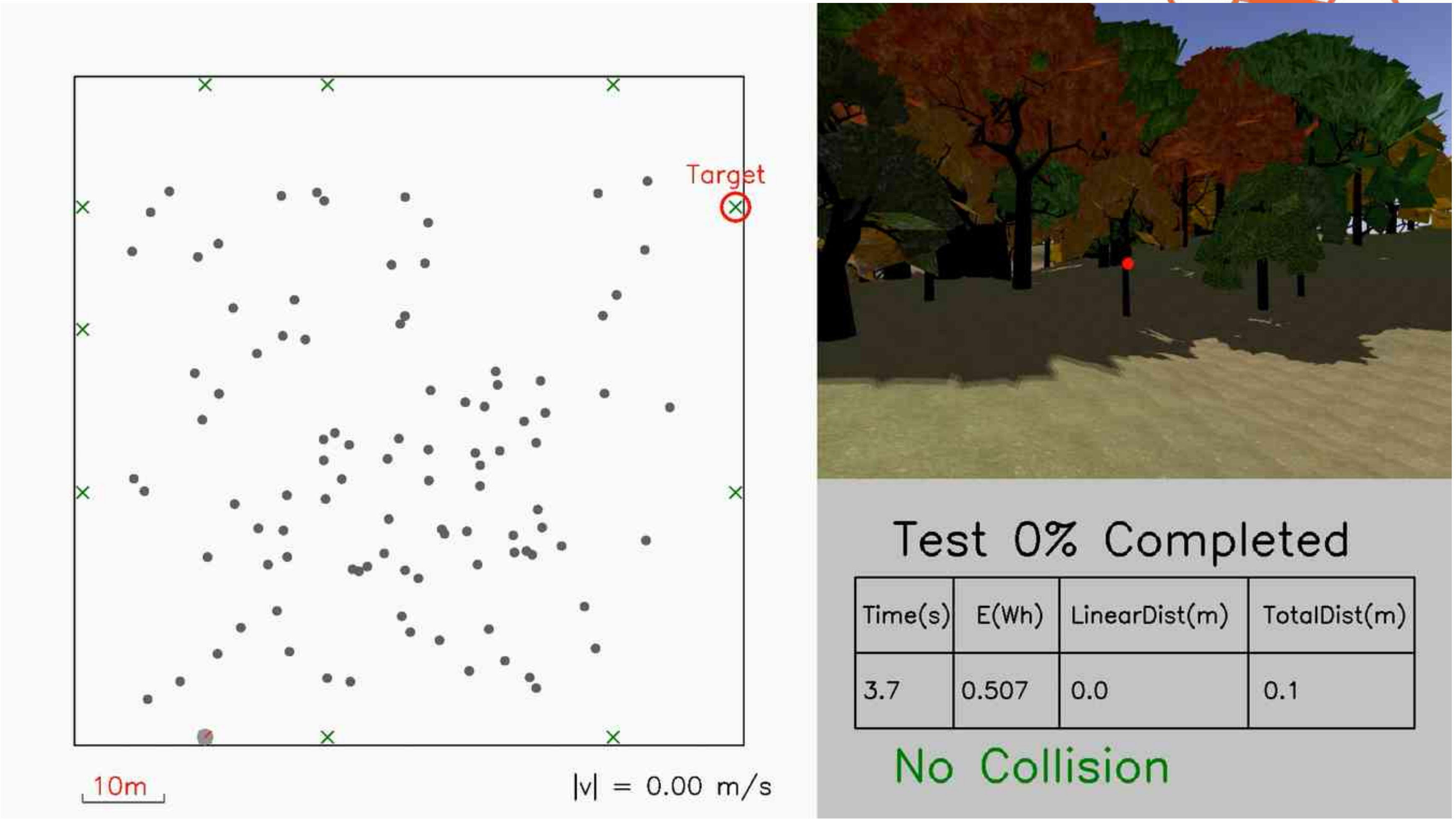
Conclusions

First step : Compatibility

- On the geometric worlds
- The tested algorithm needs to be interfaced with ROS
- To do :
  - Place launch files in a special directory and rename them
- To check and/or modify :
  - Frames
  - Inputs-Outputs topics
- To choose :
  - Control Modality

Second step : Visual Assessment

- On the forest
- Execution of a Unit test
- Automatic video generation



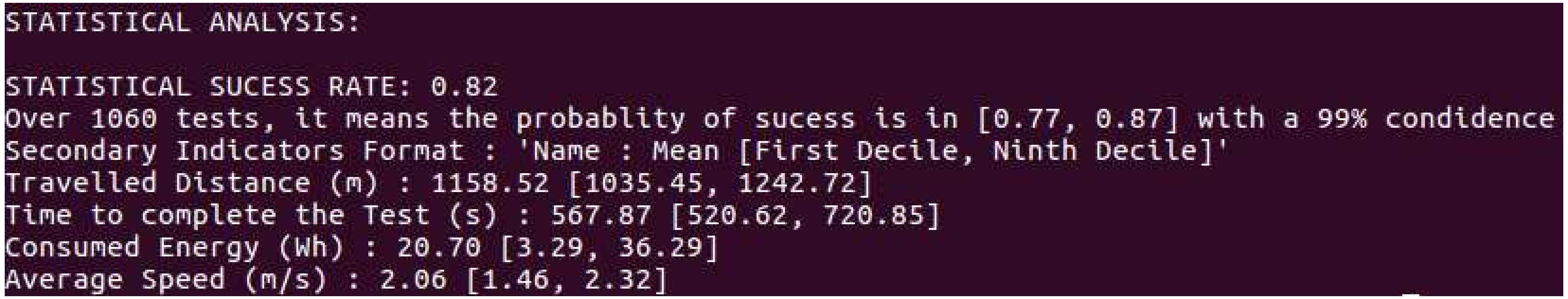
Test 0% Completed

Time(s)	E(Wh)	LinearDist(m)	TotalDist(m)
3.7	0.507	0.0	0.1

No Collision

Third step : Statistical Analysis

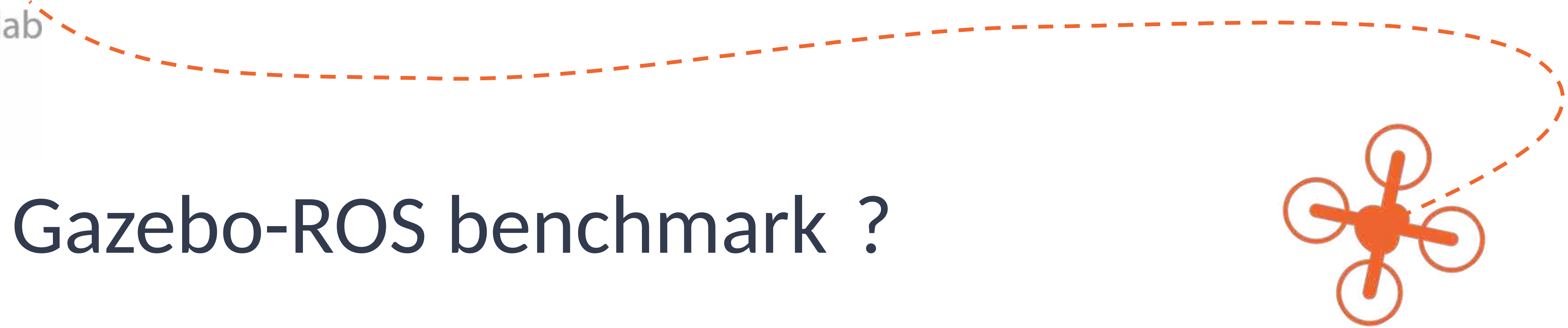
- Everything handled in a single bash script
  - Detect RotorS/Gazebo crashes and restart a test when it happens
  - Compute the indicators using a python script



```
STATISTICAL ANALYSIS:

STATISTICAL SUCESS RATE: 0.82
Over 1060 tests, it means the probablity of sucess is in [0.77, 0.87] with a 99% condidence
Secondary Indicators Format : 'Name : Mean [First Decile, Ninth Decile]'
Travelled Distance (m) : 1158.52 [1035.45, 1242.72]
Time to complete the Test (s) : 567.87 [520.62, 720.85]
Consumed Energy (Wh) : 20.70 [3.29, 36.29]
Average Speed (m/s) : 2.06 [1.46, 2.32]
```

- It takes around a week to complete the 1060 tests



A Gazebo-ROS benchmark ?

The BOARR Benchmark

Step-by-step use

Conclusions

Main features recap

- BOARR : A quadrotor obstacle avoidance benchmark :
  - Based on ROS, RotorS and Gazebo
  - Automatic Forest Generator
  - Realistic Noise on Sensors (IMU, GPS, barometer and depth sensors)
  - Wind with recurrent gust
  - Multiple modes : single test and statistical analysis
  - Scripts to start every mode using a single command line
  - Automatic video display
  - Available in open source

Going forward

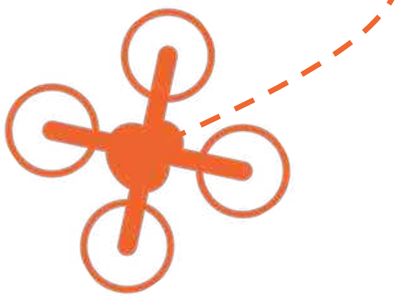
- A docker container to parallelize the statistical analysis

- Open to contributions !

  https://github.com/Gipsa-lab-PFP/BOARR

  - Improve the wind and drag effect
  - Improve the noise model of the sensors (GPS)
  - Improve the dynamics and stability of RotorS

On simulated benchmarks



- Powerful comparison tools :
  - Allow Statistical Analysis
  - Needs multiple indicators to highlight the strengths and weaknesses of all proposals
- A step toward :
  - Compatibility and Interoperability
  - Reproducibility
- Applicable to other topics

BOARR

https://github.com/Gipsa-lab-PFP/BOARR

A Benchmark for quadrotors Obstacle Avoidance using ROS and RotorS