System simulation of a fleet of drones to probe cumulus clouds

Rafael Bailon-Ruiz[1] Christophe Reymann[1] Simon Lacroix[1]
Gautier Hattenberger[2] Hector Garcia de Marina[2]
Fayçal Lamraoui[3]

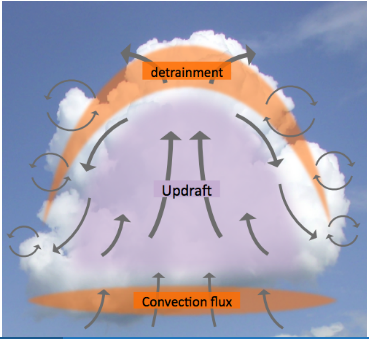
SHARC 2017

June 29, 2017

[1]LAAS-CNRS, France
[2]ENAC, France
[3]Météo-France

Context

- Characterize the boundary layer of clouds
- Follow 4D evolution of the cloud



Problem statement

Collect data with spatial resolution of 10m at 1Hz over the cloud lifespan: 1 hour over $1km^3$

Exploring clouds is a particularly complex task:

- Follow the 4D evolution of the cloud along 1D manifolds
- Highly constrained problem: Mission duration, UAV size and wind influence

The only way is to use muliple UAVs

Problem statement

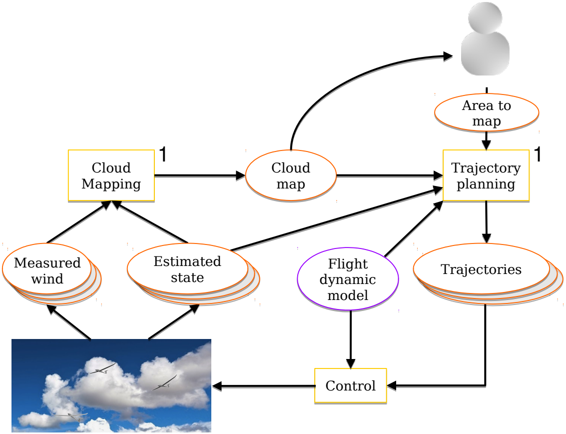
Collect data with spatial resolution of 10m at 1Hz over the cloud lifespan: 1 hour over $1km^3$

Exploring clouds is a particularly complex task:

- Follow the 4D evolution of the cloud along 1D manifolds
- Highly constrained problem: Mission duration, UAV size and wind influence

The only way is to use muliple UAVs

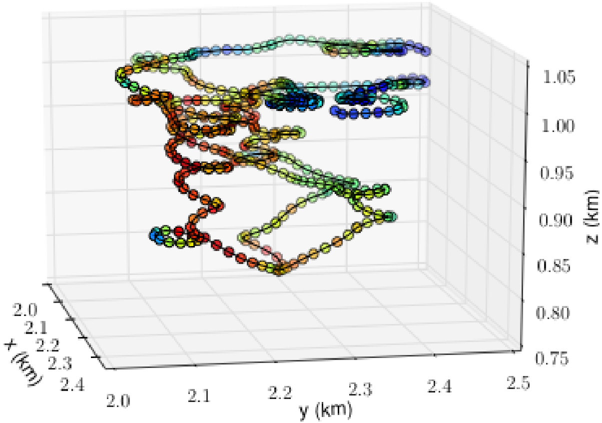
Overall structure



[1] C. Reymann, A. Renzaglia, F. Lamraoui, M. Bronz, and S. Lacroix, "Adaptive sampling of cumulus clouds with a fleet of UAVs," Autonomous robots, Jan. 2017.

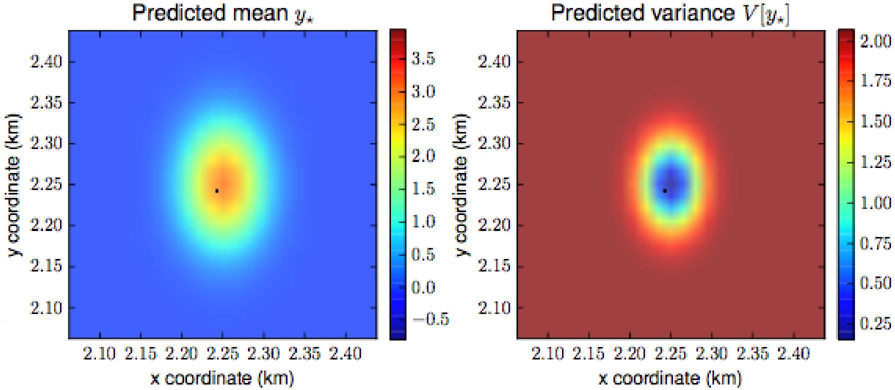
Main algorithms

Planning *Coarse level:* "map those volumes"
Fine level: on-line optimal path generation.



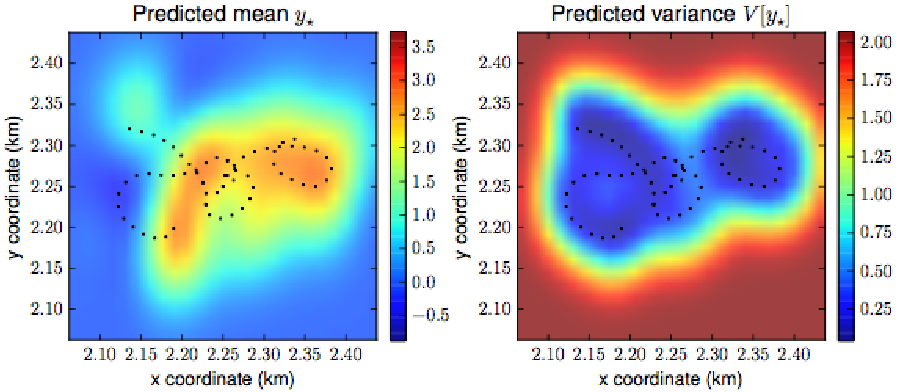
Main algorithms

Mapping Gaussian Process Regression to model the cloud



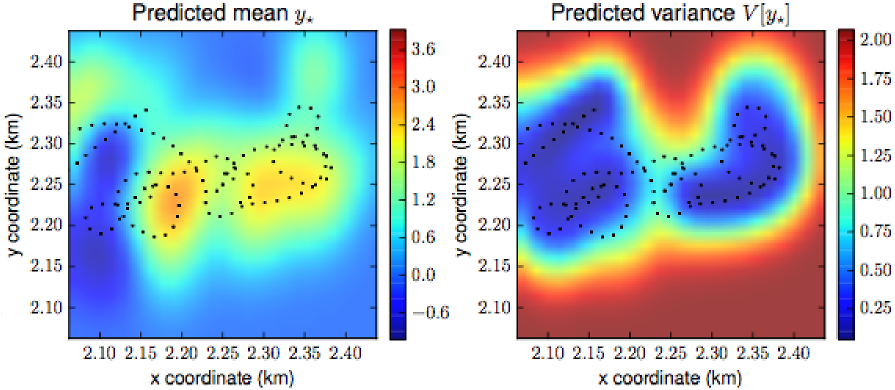
Main algorithms

Mapping Gaussian Process Regression to model the cloud



Main algorithms

Mapping Gaussian Process Regression to model the cloud



The need for a simulation architecture

Mapping and path planning were yet to be integrated with realistic flight simulators

No existing straightforward way to simulate this kind of system as a whole

Primary goals of this work:

- Build a software architecture integrating realistic simulators
- Integrate the mapping and exploration algorithms within this architecture
- Test and validate the whole system.

The need for a simulation architecture

Mapping and path planning were yet to be integrated with realistic flight simulators

No existing straightforward way to simulate this kind of system as a whole

Primary goals of this work:

- Build a software architecture integrating realistic simulators
- Integrate the mapping and exploration algorithms within this architecture
- Test and validate the whole system.

The need for a simulation architecture

Mapping and path planning were yet to be integrated with realistic flight simulators

No existing straightforward way to simulate this kind of system as a whole

Primary goals of this work:

- Build a software architecture integrating realistic simulators
- Integrate the mapping and exploration algorithms within this architecture
- Test and validate the whole system.

Simulation backends

Environment:

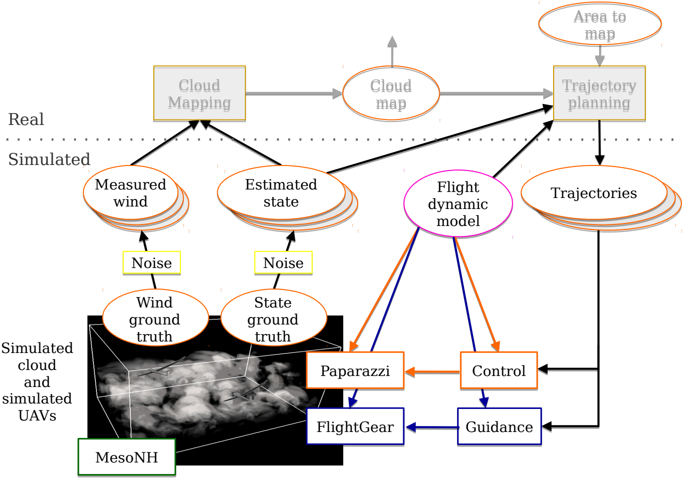
MesoNH MeteoFrance's realistic cloud simulator
Offline generation of a 64 km^3 fair weather scenario

Flight:

Paparazzi ENAC's open source autopilot and ground control software

FlightGear Open source flight simulator

Concept of the simulation architecture



Requirements of the new simulation architecture

Design a new software architecture being able to:

- Be prepared to handle a fleet of aircraft
- Integrate the project's previous work
- Seamless transfer the algorithms to the real implementation
- Add new functionality easily

We decided to use the Robot Operating System (ROS) framework

Requirements of the new simulation architecture

Design a new software architecture being able to:

- Be prepared to handle a fleet of aircraft
- Integrate the project's previous work
- Seamless transfer the algorithms to the real implementation
- Add new functionality easily

We decided to use the Robot Operating System (ROS) framework

Requirements of the new simulation architecture

Design a new software architecture being able to:

- Be prepared to handle a fleet of aircraft
- Integrate the project's previous work
- Seamless transfer the algorithms to the real implementation
- Add new functionality easily

We decided to use the Robot Operating System (ROS) framework

Requirements of the new simulation architecture

Design a new software architecture being able to:

- Be prepared to handle a fleet of aircraft
- Integrate the project's previous work
- Seamless transfer the algorithms to the real implementation
- Add new functionality easily

We decided to use the Robot Operating System (ROS) framework

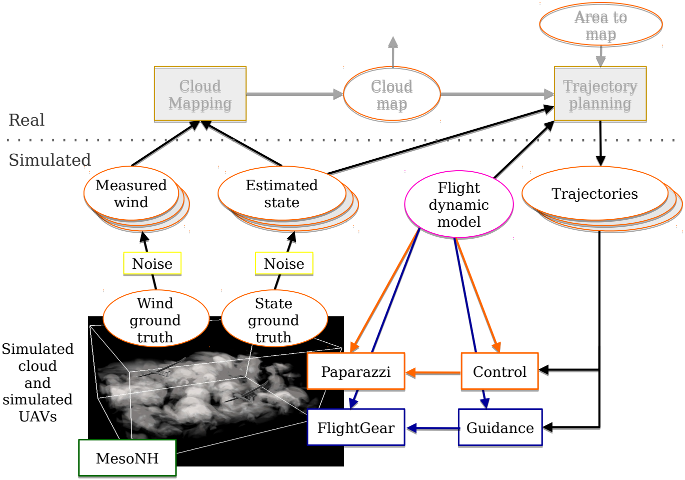
Requirements of the new simulation architecture

Design a new software architecture being able to:

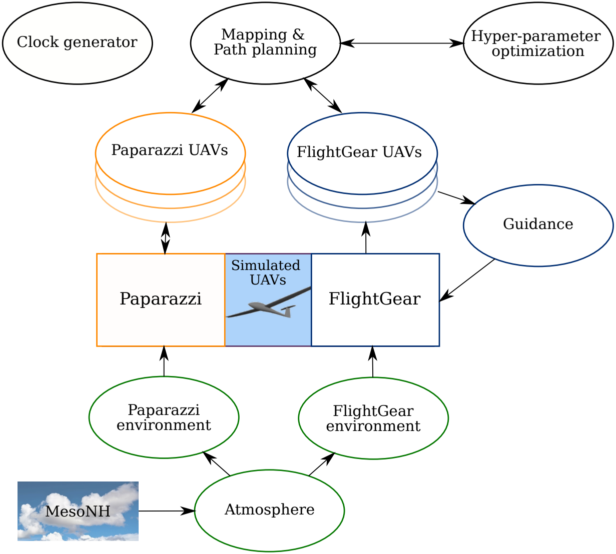
- Be prepared to handle a fleet of aircraft
- Integrate the project's previous work
- Seamless transfer the algorithms to the real implementation
- Add new functionality easily

We decided to use the Robot Operating System (ROS) framework

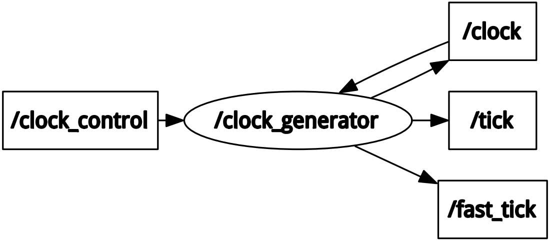
ROS implementation of the simulation architecture



ROS implementation of the simulation architecture

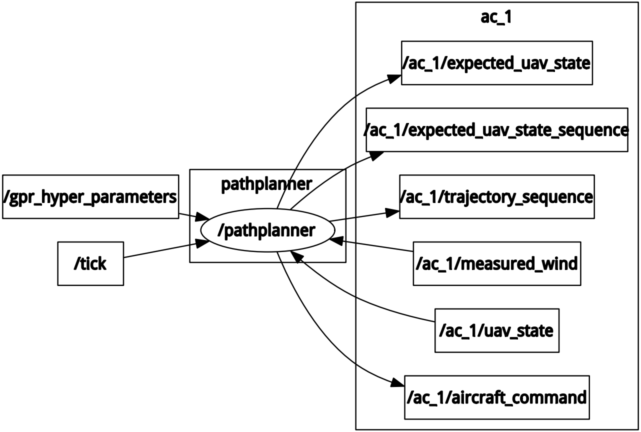


Time management



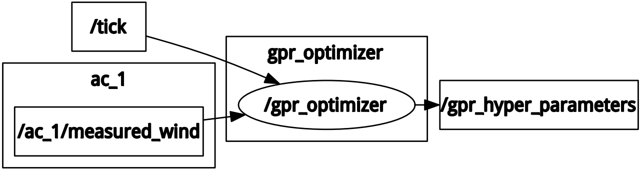
Input (Topic) Process (Node) Outputs (Topics)

Mapping & path planning node

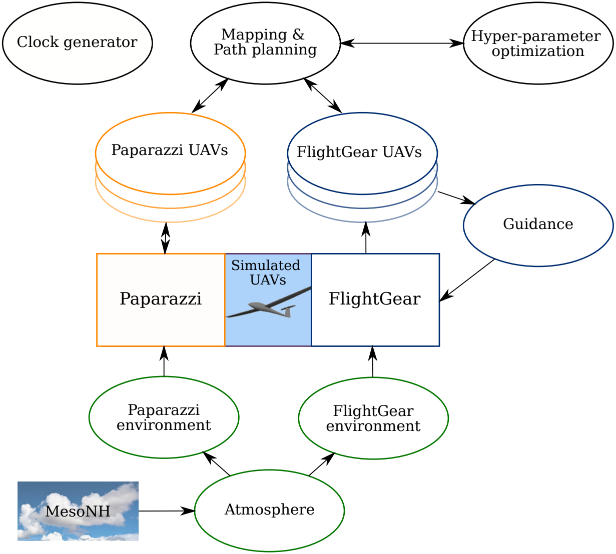


Gaussian process hyper-parameters optimization

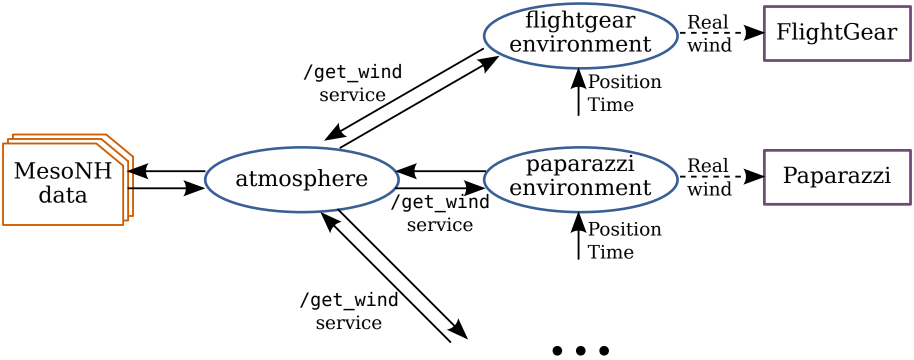
- Improve prediction with increasing wind samples



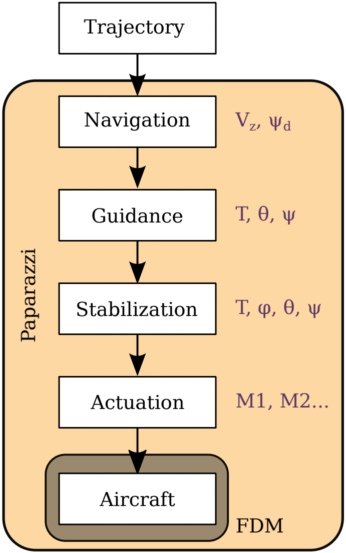
ROS implementation of the simulation architecture

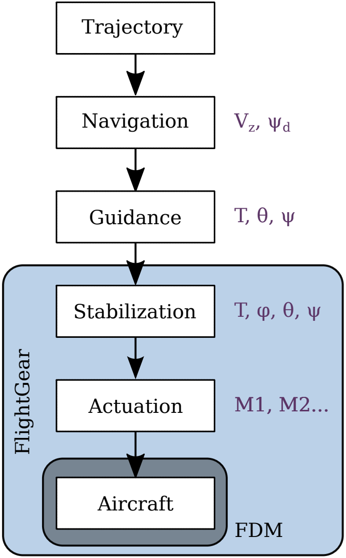


MesoNH interface with flight backends



Paparazzi vs. FlightGear control loops





Interface with Paparazzi

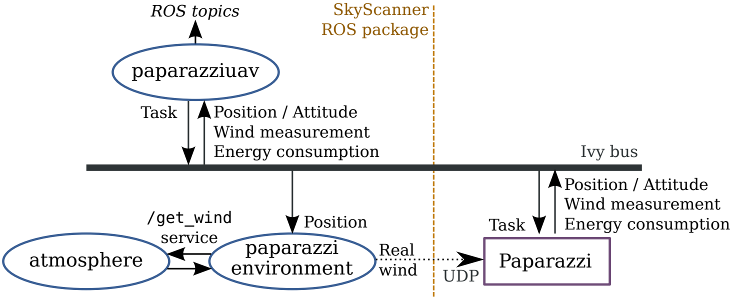


Figure: Interface between Paparazzi and SkyScanner ROS package

Interface with FlightGear

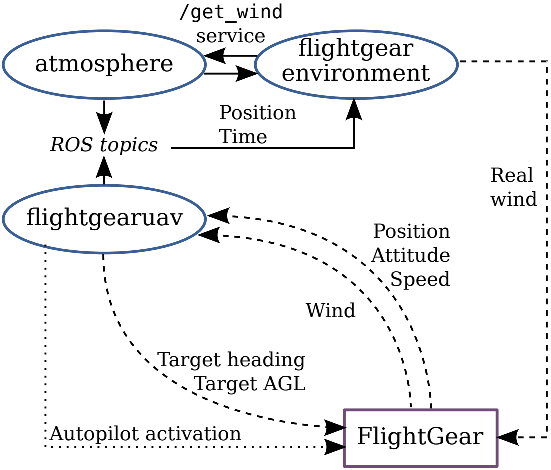


Figure: Interface scheme between FlightGear and SkyScanner ROS package

The SkyScanner ROS package

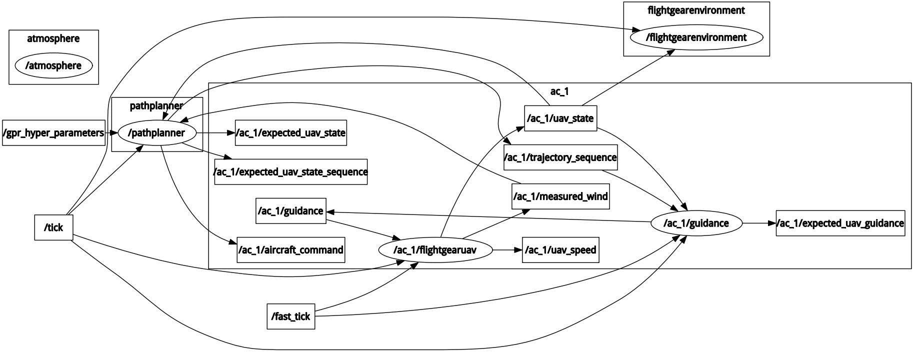
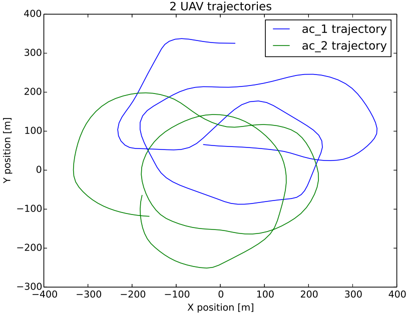
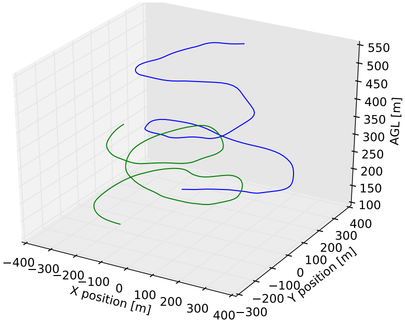


Figure: Whole simulation and control loop

Resulting trajectories





Summary

- Deployment of a simulation architecture
  - Path planning and mapping algorithms integration
  - Interfaces with realistic simulators
  - **Extensible & Reusable**
- Available in: `http://github.com/rafael1193/skyscannner_integration`