Trajectory Monitoring For A Drone Using Interval Analysis

Sylvain Largent and Julien Alexandre dit Sandretto

Department U2IS

ENSTA Paris

PPNIV'22

ENSTA

IP PARIS

Context

Properties on **dynamical systems** (mainly in robotics):

Verification

- Formal proof of safety
- Viability
- Reachability
- Limit cycle

Synthesis

- Parameter identification
- Control
- Optimization
- Motion Planning

Dynamical systems

- Dynamics $\Rightarrow$ Differential equations (ODEs or DAEs)
- Properties $\Rightarrow$ Constraints

How to guarantee safety for a UAV ?

Mainly three techniques to avoid static obstacles

- control architecture (detection of obstacles and reactive control)
- safe by construction motion planner
- control synthesis (eg. vector field)

But...

Dynamic obstacles, uncertainties, etc.

Anticipative based techniques

A realistic model can be simulated to estimate the future and detect collision (and optimize): Model predictive control for example

Monitoring

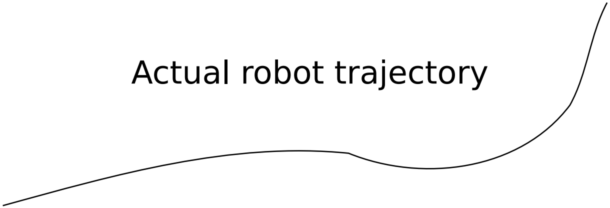
We propose a monitor that check the safety on a sliding horizon.

Realistic model ?

Very hard to obtain, in particular for a drone...

Interval analysis can help

Parameters and uncertainties (such as model discrepancies) in intervals:
$\rho \in [\rho]$
Not a trajectory but a (continuous) set of trajectories

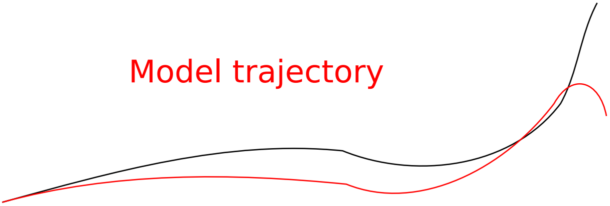


Realistic model ?

Very hard to obtain, in particular for a drone...

Interval analysis can help

Parameters and uncertainties (such as model discrepancies) in intervals: $\rho \in [\rho]$
Not a trajectory but a (continuous) set of trajectories

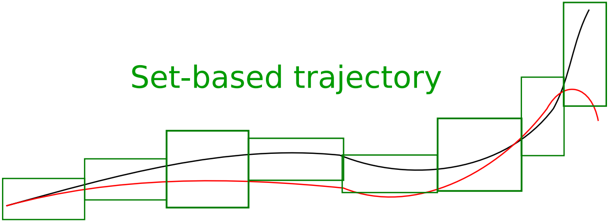


Realistic model ?

Very hard to obtain, in particular for a drone...

Interval analysis can help

Parameters and uncertainties (such as model discrepancies) in intervals:
$\rho \in [\rho]$
Not a trajectory but a (continuous) set of trajectories



Set-based model

Compromise is needed between safety and precision: "large" model is safe but useless, "thin" model is maybe too optimistic.

$\Rightarrow$ Need an identification step.

Validated simulation

Simulation of ordinary differential equations with sets, provides a tube (a list of boxes) by the help of Taylor series or Runge-Kutta (several tools: VNode, CAPD, JuliaReach, DynIbex, etc).

Parameter identification

First, choose the level of complexity for the model.

Our model

Drone is controlled by velocity: an ordinary differential equation, strongly simplified to compute quickly and monitoring online.

$$\begin{cases} \dot{x} = [\alpha_x] * u_x + [\beta_x] \\ \dot{y} = [\alpha_y] * u_y + [\beta_y] \\ \dot{z} = [\alpha_z] * u_z + [\beta_z] \end{cases} \text{or } \dot{\mathbf{w}} = [\alpha] * \mathbf{u} + [\beta] \quad (1)$$

with $u_.$ defined by a proportional controller (trajectory tracking), α and β intervals taking into account uncertainties and simplifications.

Parameter identification

Accuracy of model depends on horizon

We want actual states included in computed tube, and tube as thin as possible.

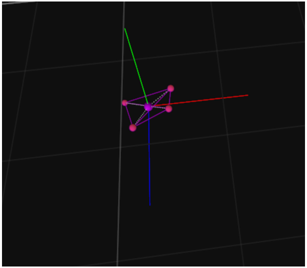
Few flights and tests to obtain values for α and β (for 1s horizon):

$$\left\{ [\alpha] * \mathbf{u} + [\beta] = \begin{pmatrix} [0,1] \\ [-0.4,0.4] \\ [-0.4,0.4] \end{pmatrix} * \mathbf{u} + \begin{pmatrix} [-0.1,0.1] \\ [-0.1,0.1] \\ [-0.1,0.1] \end{pmatrix} \right. \quad (2)$$

(More details in the paper...)

Experimentation setup



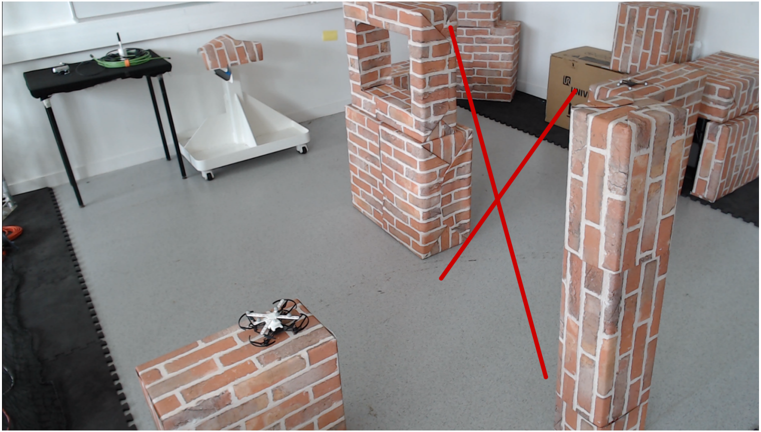



The environment

Some buildings, take-off and landing areas and a virtual door.

The environment

Some buildings, take-off and landing areas and a virtual door.



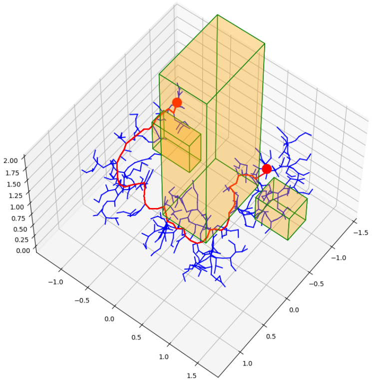
Algorithm

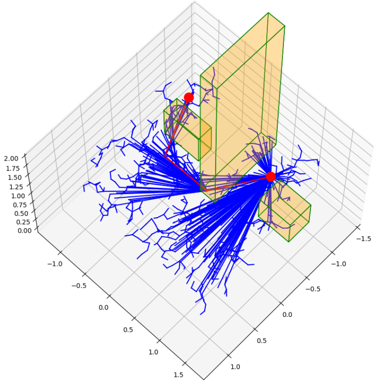
1. Plan a trajectory avoiding obstacles (enclosed in boxes) from current position to goal (with RRT*);
2. Check on a short horizon if there is a collision (with interval based monitor);
3. If no collision, let the UAV continues its trajectory; Otherwise, add or inflate previous detected obstacles to take into account uncertainties and GOTO 1;
4. At the end of the horizon, measure current position and GOTO 2.

Now: 2 and 3 are offline and simulated. Coming soon.

Algorithm

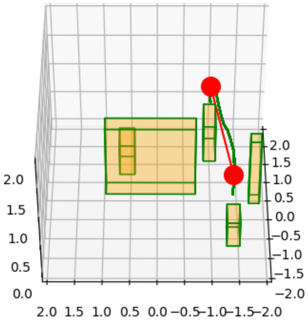
Rapidly-exploring Random Trees *RRT* and its upgraded version *RRT**

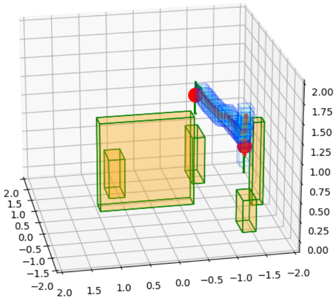




Results on two scenarios

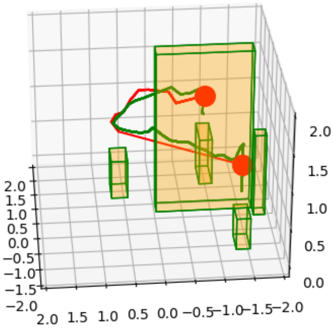
First scenario: open door

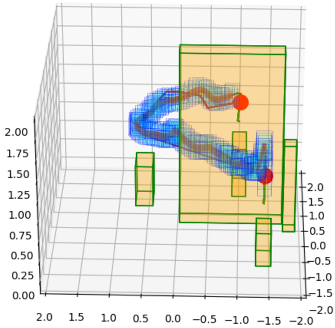




Results on two scenarios

Second scenario: closed door





Conclusion

- Interval based model of a drone, verified by experiment
- Monitoring for safety (over sliding horizon)
- The drone flies !

Future

Add the replan part in the global architecture, the detection of dynamical obstacles, etc.

Questions ?