Validated non collision prediction of multiple drones

AID Meeting

Julien Alexandre dit Sandretto   Alexandre Chapoutot   Christophe Garion
Olivier Mullier   Xavier Thirioux   **Ghilès Ziat**

March, 13, 2020

Context: Swarm of Drones

Safety of an evolving swarm of drones in term of collision avoidance

Two types of obstacles:

- static: forbidden areas;
- dynamic: the robots themselves

Goal

Guarantee on the non collision of the swarm with the environment (static obstacles) and the other drones (dynamic obstacles).

A drone i from the swarm is modeled with a controlled dynamical system:

$$(\mathcal{S}_i) \begin{cases} \dot{\mathbf{x}}_i = f(\mathbf{x}_i, \mathbf{u}_i) \\ \mathbf{x}_i(0) = \mathbf{x}_{i;0} \end{cases}$$

From a given control $\mathbf{u}_i$ and a given time horizon T on which the control is applied to the system, the goal is to prove that, for two drones i and j:

$$x_i(t) \neq x_j(t), \forall t \in [0, T] \tag{1}$$

Uncertainties make this constraint intractable to check in general.

Validated method

Use of outer approximations to guarantee the non collision.

Interval Analysis

An interval is denoted $[x] = [\underline{x}, \overline{x}]$ with $\underline{x} \leqslant \overline{x}$.

The set of intervals is denoted

$$\mathbb{IR} = \{[x] = [\underline{x}, \overline{x}] \mid \underline{x}, \overline{x} \in \mathbb{R},\ \underline{x} \leqslant \overline{x}\}.$$

The Cartesian product of intervals $[\mathbf{x}] \in \mathbb{IR}^n$ is a box.

Interval arithmetic

Evaluating an arithmetic expression with intervals leads to an outer-approximation of the set it defines.

To deal with interval functions, an interval inclusion function (or interval extension) of a function can be defined.

Examples:

- natural extension: replaces the operations on reals by their interval counterparts using interval arithmetic;
- mean value extension: linearizes the function around its mean value.

Definition (IVP-ode)

An IVP-ODE is defined as

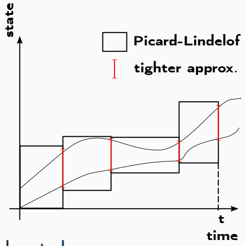
$$(\mathcal{S}) \begin{cases} \dot{\mathbf{y}} = f(t, \mathbf{y}) \\ \mathbf{y}(0) \in \mathcal{Y}_0 \subseteq \mathbb{R}^n, \; t \in [0, t_{\text{end}}] \end{cases} .$$

Goal is to compute $\mathbf{y}(t; \mathcal{Y}_0) = \{\mathbf{y}(t; \mathbf{y}_0) \mid \mathbf{y}_0 \in \mathcal{Y}_0\}$.

Phase 1 a priori enclosure $\left[\mathbf{y}_{[t_i, t_{i+1}]}\right]$ of

$$\{\mathbf{y}(t_k; \mathbf{y}_i) \mid t_k \in [t_i, t_{i+1}], \mathbf{y}_i \in [\mathbf{y}_i]\}$$

Phase 2 tight enclosure $[\mathbf{y}_{t_{i+1}}]$ at time t_{i+1}.



A trajectory then consists in a set of boxes called a tube.

If we consider again the dynamic of a drone i:

$$(\mathcal{S}_i)\begin{cases}\dot{\mathbf{x}}_i = f(\mathbf{x}_i, \mathbf{u}_i)\\ \mathbf{x}_i(0) \in [\mathbf{x}_{i;0}]\\ \mathbf{u}_i \in [\mathbf{u}_i]\end{cases}$$

The control $\mathbf{u}_i$ is considered constant during the simulation over time t.

we infer a solution operator

$$\mathbf{x}_i(t, [\mathbf{x}_{i,0}], [\mathbf{u}_i]) = \{\mathbf{x}(t; \mathbf{x}_0; \mathbf{u}_i) \mid \mathbf{x}_0 \in [\mathbf{x}_{i;0}], \mathbf{u}_i \in [\mathbf{u}_i]\}.$$

and its associated interval inclusion $[\mathbf{x}_i](t, [\mathbf{x}_{i,0}], [\mathbf{u}_i])$.

DynIBEX

Validated Simulation with Runge-Kutta

- Proof of existence and uniqueness of solution for ODEs and DAEs,
- Local truncation error computation for any Runge-Kutta method (implicit or explicit),
- Combined with contractors (HC4, *etc.*).

Verification of temporal constraints

- Stayed in $\mathcal{A}$ until $\tilde{t} < t_{\text{end}}$:

$$\forall t \in [0, \tilde{t}]\,,\ \{\mathbf{y}(t; \mathbf{y}_0) \mid \mathbf{y}_0 \in [\mathbf{y}_0]\} \subseteq \text{int}(\mathcal{A})$$

- Included in $\mathcal{A}$ inside $[0, t_{\text{end}}]$:

$$\exists t \in [0, t_{\text{end}}]\,,\ \{\mathbf{y}(t; y_0) \mid \mathbf{y}_0 \in [\mathbf{y}_0]\} \subseteq \text{int}(\mathcal{A}).$$

Solving the Problem with DynIBEX

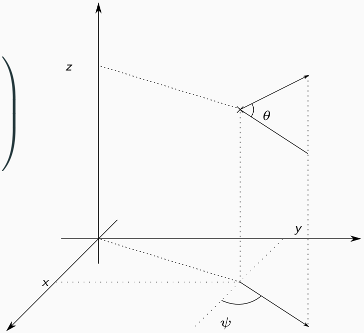
Temporal constraint

Has crossed $\mathcal{A}$ (before τ):

$$\exists t < \tau, \mathbf{y}(t) \cap \Box\mathcal{A} \neq \emptyset$$

We can define the collision detection in term of this temporal constraint.

$$(\mathcal{S}_i)\begin{cases}\dot{\mathbf{X}}_i = \begin{pmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{z}_i\end{pmatrix} = \begin{pmatrix} v_i\cos\psi_i\cos\theta_i\\ v\sin\psi_i\cos\theta_i\\ v\sin\theta_i\end{pmatrix}\\ \mathbf{X}_i(0)\in[\mathbf{X}_0]\end{cases}$$



with

- the state vector $\mathbf{X}_i = (x_i, y_i, z_i)^T$ representing the position of the robot;
- the control vector $\mathbf{u}_i = (v_i, \psi_i, \theta_i)^T$ consisting in the velocity v_i, the heading angle ψ_i and the track angle θ_i.

drone ($\mathcal{S}_1$):

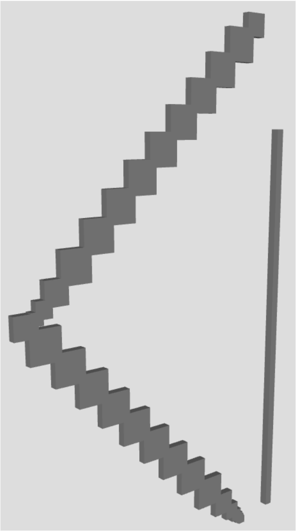
- $\mathbf{X}_1 = (1, 1, 1)$ and $v_1 = 1$;
- $\psi_1 = \frac{\pi}{2}$ and $\theta_1 = \frac{\pi}{2}$.

drone ($\mathcal{S}_2$):

- $\mathbf{X}_2 = (1, 7.8, 7.8)$ and $v_2 = -1$;
- $\psi_2 = -\frac{\pi}{2}$ and $\theta_2 = -\frac{3\pi}{4}$.

drone ($\mathcal{S}_3$):

- $\mathbf{X}_3 = (0, 1, 2)$ and $v_3 = 1$;
- $\psi_3 = \pi$ and $\theta_3 = \frac{\pi}{2}$.



The simulation time is 10s.

For N drones, we obtain the N tubes :

- $(\mathcal{S}_1)$: $\left\{ \left[\mathbf{x}_{1;\left[t_{\mathbf{1},\mathbf{0}}\right]}\right], \left[\mathbf{x}_{1;\left[t_{\mathbf{1},\mathbf{1}}\right]}\right], \dots, \left[\mathbf{x}_{1;\left[t_{\mathbf{1},m_{\mathbf{1}}}\right]}\right] \right\}$
- $(\mathcal{S}_2)$: $\left\{ \left[\mathbf{x}_{2;\left[t_{\mathbf{2},\mathbf{0}}\right]}\right], \left[\mathbf{x}_{2;\left[t_{\mathbf{2},\mathbf{1}}\right]}\right], \dots, \left[\mathbf{x}_{2;\left[t_{\mathbf{2},m_{\mathbf{2}}}\right]}\right] \right\}$
- $\vdots$
- $(\mathcal{S}_N)$: $\left\{ \left[\mathbf{x}_{N;\left[t_{N,\mathbf{0}}\right]}\right], \left[\mathbf{x}_{N;\left[t_{N,\mathbf{1}}\right]}\right], \dots, \left[\mathbf{x}_{N;\left[t_{N,m_N}\right]}\right] \right\}$

with $\left[\mathbf{x}_{i;\left[t_{i,j}\right]}\right] \supseteq \{\mathbf{x}_i(t) \mid t \in [t_{i,j}]\}$

Algorithm 1: Algorithm for checking the non collision between a set of drones.

Input: $(\mathcal{S}_1) : \left\{ \left[\mathbf{x}_{1;\left[t_{\mathbf{1},\mathbf{0}}\right]}\right], \left[\mathbf{x}_{1;\left[t_{\mathbf{1},\mathbf{1}}\right]}\right], \ldots, \left[\mathbf{x}_{1;\left[t_{\mathbf{1},m_{\mathbf{1}}}\right]}\right] \right\}$

Input: $(\mathcal{S}_2) : \left\{ \left[\mathbf{x}_{2;\left[t_{\mathbf{2},\mathbf{0}}\right]}\right], \left[\mathbf{x}_{2;\left[t_{\mathbf{2},\mathbf{1}}\right]}\right], \ldots, \left[\mathbf{x}_{2;\left[t_{\mathbf{2},m_{\mathbf{2}}}\right]}\right] \right\}$

$\vdots$

Input: $(\mathcal{S}_N) : \left\{ \left[\mathbf{x}_{N;\left[t_{N,\mathbf{0}}\right]}\right], \left[\mathbf{x}_{N;\left[t_{N,\mathbf{1}}\right]}\right], \ldots, \left[\mathbf{x}_{N;\left[t_{N,m_N}\right]}\right] \right\}$

for $i = 1$ *to* N **do**
  for $j = 0$ *to* m_i **do**
    for $k = i + 1$ *to* N **do**
      for $l = 0$ *to* m_k **do**
        if $[t_{i;j}] \cap [t_{k;l}] \neq \emptyset$ **then**
          if $\left[\mathbf{x}_{i;\left[t_{i,j}\right]}\right] \cap \left[\mathbf{x}_{k;\left[t_{k,l}\right]}\right] \neq \emptyset$ **then**
            possible collision

no collision detected

We end up with a solution with a high complexity.

DynIbex

- handle the dynamics;
- handle static obstacles constraints easily (for example boxRRT);
- requires more sophisticated tools to handle the dynamic obstacles.

A solution:

AbSolute

(Constraint solver based on abstract domains)

From DynIBEX to AbSolute

We cast the results from DynIBEX into a Constraint Satisfaction Problem.

init

real $x_0 = [-10000.000000; 10000.000000]$
real $x_1 = [-10000.000000; 10000.000000]$
real $x_2 = [-10000.000000; 10000.000000]$
real $t = [0; 100.0]$

constraints

T_0 :

$$\left(t \text{ in } [t_{0,0}] \,\&\&\, x_0 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{0},\mathbf{0}}]}\right]_0 \,\&\&\, x_1 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{0},\mathbf{0}}]}\right]_1 \,\&\&\, x_2 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{0},\mathbf{0}}]}\right]_2\right) ||$$

$$\left(t \text{ in } [t_{0,1}] \,\&\&\, x_0 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{0},\mathbf{1}}]}\right]_0 \,\&\&\, x_1 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{0},\mathbf{1}}]}\right]_1 \,\&\&\, x_2 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{0},\mathbf{1}}]}\right]_2\right) ||$$

$\vdots$

T_1 :

$$\left(t \text{ in } [t_{1,0}] \,\&\&\, x_0 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{1},\mathbf{0}}]}\right]_0 \,\&\&\, x_1 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{1},\mathbf{0}}]}\right]_1 \,\&\&\, x_2 \text{ in } \left[\mathbf{x}_{1;[t_{\mathbf{1},\mathbf{0}}]}\right]_2\right) ||$$