Distributed localization with intervals

L. Jaulin
ENSTA Bretagne, LabSTICC
Présentation Thales
May 16, Brest
Interval analysis
Problem. Given \( f : \mathbb{R}^n \to \mathbb{R} \) and a box \([x] \subset \mathbb{R}^n\), prove that

\[
\forall x \in [x], \quad f(x) \geq 0.
\]

Interval arithmetic can solve efficiently this problem.
Example. Is the function

\[ f(x) = x_1 x_2 - (x_1 + x_2) \cos x_2 + \sin x_1 \cdot \sin x_2 + 2 \]

always positive for \( x_1, x_2 \in [-1, 1] \) ?
Interval arithmetic

\([-1,3] + [2,5] = ?,
\([-1,3] \cdot [2,5] = ?,
\text{abs}([-7,1]) = ?\)
Interval arithmetic

\[ [-1, 3] + [2, 5] = [1, 8], \]

\[ [-1, 3] \cdot [2, 5] = [-5, 15], \]

\[ \text{abs}([-7, 1]) = [0, 7] \]
The interval extension of

\[ f(x_1, x_2) = x_1 \cdot x_2 - (x_1 + x_2) \cdot \cos x_2 + \sin x_1 \cdot \sin x_2 + 2 \]

is

\[ [f][[x_1], [x_2]] = [x_1] \cdot [x_2] - ([x_1] + [x_2]) \cdot \cos [x_2] + \sin [x_1] \cdot \sin [x_2] + 2. \]
**Theorem** (Moore, 1970)

\[ [f][x] \subset \mathbb{R}^+ \Rightarrow \forall x \in [x], f(x) \geq 0. \]
Set Inversion
A subpaving of $\mathbb{R}^n$ is a set of non-overlapping boxes of $\mathbb{R}^n$. Compact sets $X$ can be bracketed between inner and outer subpavings:

$$X^- \subset X \subset X^+. $$
Example.

\[ \mathbf{X} = \{ (x_1, x_2) \mid x_1^2 + x_2^2 \in [1, 2] \}. \]
Let $f : \mathbb{R}^n \to \mathbb{R}^m$ and let $Y$ be a subset of $\mathbb{R}^m$. Set inversion is the characterization of

$$X = \{ x \in \mathbb{R}^n \mid f(x) \in Y \} = f^{-1}(Y).$$
We shall use the following tests.

(i) \[[f](x)\] \subset Y \Rightarrow [x] \subset X

(ii) \[[f](x)\] \cap Y = \emptyset \Rightarrow [x] \cap X = \emptyset.

Boxes for which these tests failed, will be bisected, except if they are too small.
Dynamical localization
The operator $C : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ is a *contractor* [4] for the equation $f(x) = 0$, if

$$\left\{ \begin{array}{ll}
C([x]) \subset [x] & \text{(contractance)} \\
x \in [x] \text{ and } f(x) = 0 \Rightarrow x \in C([x]) & \text{(consistence)}
\end{array} \right.$$
Building contractors
Consider the primitive equation

\[ x_1 + x_2 = x_3 \]

with \( x_1 \in [x_1], \ x_2 \in [x_2], \ x_3 \in [x_3] \).
We have

\[ x_3 = x_1 + x_2 \Rightarrow x_3 \in [x_3] \cap ([x_1] + [x_2]) \]
\[ x_1 = x_3 - x_2 \Rightarrow x_1 \in [x_1] \cap ([x_3] - [x_2]) \]
\[ x_2 = x_3 - x_1 \Rightarrow x_2 \in [x_2] \cap ([x_3] - [x_1]) \]
The contractor associated with $x_1 + x_2 = x_3$ is thus

$$\mathcal{C} \left( \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) = \left( \begin{bmatrix} x_1 \cap ([x_3] - [x_2]) \\ x_2 \cap ([x_3] - [x_1]) \\ x_3 \cap ([x_1] + [x_2]) \end{bmatrix} \right)$$
Tubes
A trajectory is a function $f : \mathbb{R} \rightarrow \mathbb{R}^n$. [6, 5]. For instance

$$f(t) = \begin{pmatrix} \cos t \\ \sin t \end{pmatrix}$$

is a trajectory.
Order relation

\[ f \leq g \iff \forall t, \forall i, f_i(t) \leq g_i(t). \]
We have

\begin{align*}
    h = f & \land g \iff \forall t, \forall i, h_i(t) = \min(f_i(t), g_i(t)), \\
    h = f & \lor g \iff \forall t, \forall i, h_i(t) = \max(f_i(t), g_i(t)).
\end{align*}
The set of trajectories is a lattice. Interval of trajectories (tubes) can be defined.
Example.

\[ [f](t) = \begin{pmatrix} \cos t + [0, t^2] \\ \sin t + [-1, 1] \end{pmatrix} \]

is an interval trajectory (or tube).
Tube arithmetics
If $[x]$ and $[y]$ are two scalar tubes [1], we have

- $[z] = [x] + [y] \Rightarrow [z](t) = [x](t) + [y](t)$ (sum)
- $[z] = \text{shift}_a([x]) \Rightarrow [z](t) = [x](t + a)$ (shift)
- $[z] = [x] \circ [y] \Rightarrow [z](t) = [x]([y](t))$ (composition)
- $[z] = \int [x] \Rightarrow [z](t) = \left[ \int_0^t x^-(\tau) d\tau, \int_0^t x^+(\tau) d\tau \right]$ (integral)
Tube Contractors
Tube arithmetic allows us to build contractors [3].
Consider for instance the differential constraint

\[ \dot{x}(t) = x(t + \tau) \cdot u(t), \]
\[ x(t) \in [x](t), \dot{x}(t) \in [\dot{x}](t), u(t) \in [u](t), \tau \in [\tau] \]

We decompose as follows

\[
\begin{cases}
  x(t) &= x(0) + \int_0^t y(\tau) \, d\tau \\
  y(t) &= a(t) \cdot u(t) \\
  a(t) &= x(t + \tau)
\end{cases}
\]
Possible contractors are

\[
\begin{align*}
[x](t) &= [x](t) \cap ([x](0) + \int_0^t [y](\tau) d\tau) \\
[y](t) &= [y](t) \cap [a](t) \cdot [u](t) \\
[u](t) &= [u](t) \cap \frac{[y](t)}{[a](t)} \\
[a](t) &= [a](t) \cap \frac{[y](t)}{[u](t)} \\
[a](t) &= [a](t) \cap [x](t + [\tau]) \\
[x](t) &= [x](t) \cap [a](t - [\tau]) \\
[\tau] &= [\tau](t) \cap \ldots
\end{align*}
\]
**Example.** Consider $x(t) \in [x](t)$ with the constraint

$$\forall t, \ x(t) = x(t + 1)$$

Contract the tube $[x](t)$. 
We first decompose into primitive trajectory constraints

\[ x(t) = a(t+1) \]
\[ x(t) = a(t). \]
Contractors

\[ [x](t) : = [x](t) \cap [a](t + 1) \]
\[ [a](t) : = [a](t) \cap [x](t - 1) \]
\[ [x](t) : = [x](t) \cap [a](t) \]
\[ [a](t) : = [a](t) \cap [x](t) \]
Interval analysis
Dynamical localization
Swarm localization

Distributed localization with intervals
Interval analysis
Dynamical localization
Swarm localization
Distributed localization with intervals
Interval analysis
Dynamical localization
Swarm localization
Distributed localization with intervals
Definition

A tube $[x](\cdot)$ is defined as an envelope enclosing an uncertain trajectory $x(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^n$. It is built as an interval of two functions $[x^{-}(\cdot), x^{+}(\cdot)]$ such that $\forall t, x^{-}(t) \leq x^{+}(t)$. A trajectory $x(\cdot)$ belongs to the tube $[x](\cdot)$ if $\forall t, x(t) \in [x](t)$. Fig. 1 illustrates a tube implemented with a set of boxes. This sliced implementation is detailed hereinafter.

Fig. 1 A tube $[x](\cdot)$ represented by a set of slices. This representation can be used to enclose signals such as $x^{\ast}(\cdot)$.

Code example:

```plaintext
float timestep = 0.1;
interval domain(0, 1);
Tubex x(domain, timestep, Function("t^2", (t-5)^2 + [-0.5,0.5]));
```
Time-space estimation
Classical state estimation

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \quad t \in \mathbb{R} \\
0 &= g(x(t), t) \quad t \in T \subset \mathbb{R}.
\end{align*}
\]

Space constraint \( g(x(t), t) = 0 \).
Example.

\[ \begin{align*}
\dot{x}_1 &= x_3 \cos x_4 \\
\dot{x}_2 &= x_3 \cos x_4 \\
\dot{x}_3 &= u_1 \\
\dot{x}_4 &= u_2 \\
(x_1 (5) - 1)^2 + (x_2 (5) - 2)^2 - 4 &= 0 \\
(x_1 (7) - 1)^2 + (x_2 (7) - 2)^2 - 9 &= 0
\end{align*} \]
With time-space constraints

\[
\begin{cases}
\dot{x}(t) &= f(x(t), u(t)) & t \in \mathbb{R} \\
0 &= g(x(t), x(t'), t, t') & (t, t') \in T \subset \mathbb{R} \times \mathbb{R}.
\end{cases}
\]
Example. An ultrasonic underwater robot with state

\[ x = (x_1, x_2, \ldots) = (x, y, \theta, v, \ldots) \]

At time \( t \) the robot emits an omnidirectional sound. At time \( t' \) it receives it

\[ (x_1 - x'_1)^2 + (x_2 - x'_2)^2 - c (t - t')^2 = 0. \]
Swarm localization
Consider $n$ robots $R_1, \ldots, R_n$ described by

$$\dot{x}_i = f(x_i, u_i), u_i \in [u_i].$$
Omnidirectional sounds are emitted and received. A *ping* is a 4-uple \((a, b, i, j)\) where \(a\) is the emission time, \(b\) is the reception time, \(i\) is the emitting robot and \(j\) the receiver.
Interval analysis
Dynamical localization
Swarm localization

Distributed localization with intervals
With the time space constraint

\[ \dot{x}_i = f(x_i, u_i), u_i \in [u_i]. \]

\[ g(x_{i(k)}(a(k)), x_{j(k)}(b(k)), a(k), b(k)) = 0 \]

where

\[ g(x_i, x_j, a, b) = \|x_1 - x_2\| - c(b - a). \]
Clocks are uncertain. We only have measurements \( \tilde{a}(k), \tilde{b}(k) \) of \( a(k), b(k) \) thanks to clocks \( h_i \). Thus

\[
\dot{x}_i = f(x_i, u_i), u_i \in [u_i].
\]

\[
g(x_{i(k)}(a(k)), x_{j(k)}(b(k)), a(k), b(k)) = 0
\]

\[
\tilde{a}(k) = h_{i(k)}(a(k))
\]

\[
\tilde{b}(k) = h_{j(k)}(b(k))
\]
The drift of the clocks is bounded

\[ \dot{x}_i = f(x_i, u_i), u_i \in [u_i]. \]
\[ g(x_{i(k)}(a(k)), x_{j(k)}(b(k)), a(k), b(k)) = 0 \]
\[ \tilde{a}(k) = h_{i(k)}(a(k)) \]
\[ \tilde{b}(k) = h_{j(k)}(b(k)) \]
\[ \dot{h}_i = 1 + n_h, \quad n_h \in [n_h] \]
https://youtu.be/jr8xKle0Nds
https://youtu.be/GycJxGFvYE8
Interval analysis
Dynamical localization
Swarm localization

https://youtu.be/GVGTwnJ_dpQ

Distributed localization with intervals
F. Le Bars, J. Sliwka, O. Reynet, and L. Jaulin.  
State estimation with fleeting data.  

A. Bethencourt and L. Jaulin.  
Cooperative localization of underwater robots with unsynchronized clocks.  

A. Bethencourt and L. Jaulin.  
Solving non-linear constraint satisfaction problems involving time-dependant functions.  
*Mathematics in Computer Science, 8*(3), 2014.

G. Chabert and L. Jaulin.  
Contractor Programming.  

S. Rohou.
Reliable robot localization: a constraint programming approach over dynamical systems.
PhD dissertation, Université de Bretagne Occidentale, ENSTA-Bretagne, France, December 2017.