

# Overview of the *IBEX* library

**Jordan Ninin**

*Lab-STICC / ENSTA-Bretagne  
Brest, France*

with

Gilles CHABERT

*INRIA / Ecole des Mines de Nantes  
et al.*

Toulouse, March 2016

# Introduction

## Observations:

- In Automatic, Robotic, Electronic or Mechanic, engineers know very well their problems.
- They solve iteratively their problem by increasing the difficulty, the number of constraints to converge to the real problem they want to solve
- But, most of time they change the kind of optimization problem: LP, NLP, MINLP, SDP, DFO,...

# Introduction

## Observations:

- In Automatic, Robotic, Electronic or Mechanic, engineers know very well their problems.
- They solve iteratively their problem by increasing the difficulty, the number of constraints to converge to the real problem they want to solve
- But, most of time they change the kind of optimization problem: LP, NLP, MINLP, SDP, DFO,...

### What do they need?

- ⇒ **Involve** with the modelization improvement.
- ⇒ Construct the best optimizer for their **own** problems.

# Introduction

## Observations:

- In Automatic, Robotic, Electronic or Mechanic, engineers know very well their problems.
- They solve iteratively their problem by increasing the difficulty, the number of constraints to converge to the real problem they want to solve
- But, most of time they change the kind of optimization problem: LP, NLP, MINLP, SDP, DFO,...

## What do they need?

- ⇒ **Involve** with the modelization improvement.
- ⇒ Construct the best optimizer for their **own** problems.

# IBEX

# Contents

## ① Concepts

Set-membership approach

Contractor

Properties

Implementation

## ② IBEXopt

Constraint Satisfaction Problem

Global Optimisation

## ③ Related Project

DynIBEX

ViabIBEX

$H_\infty$  control synthesis

# Contents

## ① Concepts

Set-membership approach

Contractor

Properties

Implementation

## ② IBEXopt

Constraint Satisfaction Problem

Global Optimisation

## ③ Related Project

DynIBEX

ViabIBEX

$H_\infty$  control synthesis

# Interval Arithmetic

## Ideas

- Consider Sets in place of Real Numbers
- Need ordered structure: Lattice

$$\mathbb{IR} = \{[\underline{x}, \bar{x}] : \underline{x} \in \mathbb{R} \cup \{-\infty\}, \bar{x} \in \mathbb{R} \cup \{+\infty\} \text{ and } \underline{x} \leq \bar{x}\} \cup \{\emptyset\}$$

# Interval Arithmetic

## Ideas

- Consider Sets in place of Real Numbers
- Need ordered structure: Lattice

$$\mathbb{IR} = \{[\underline{x}, \bar{x}] : \underline{x} \in \mathbb{R} \cup \{-\infty\}, \bar{x} \in \mathbb{R} \cup \{+\infty\} \text{ and } \underline{x} \leq \bar{x}\} \cup \{\emptyset\}$$

Why do we use the Interval Arithmetic  $\mathbb{IR}^n$ ?

- Based on about 50 years of experience and studies,
- Best way to manipulate sets and boxes,
- There is no problem to deal with discontinuity or unusual functions:
  - $\forall \mathbf{x} \in \mathbb{IR}, f(x) = 1/x$  is well defined:  $f([0, 0]) = [-\infty, \infty]$ .
  - $(\nabla abs)([0, 0]) = [-1, 1]$ ,
  - $\chi(\mathbf{x}, \mathbf{y}, \mathbf{z})$  to implement IF statement
  - $atan2(y, x)$  give the angle of the vector  $(x, y)$ .
- Easy to enclose Algorithm as **contractor**,



# Definition: Contractor

Let  $\mathbb{X} \subseteq \mathbb{R}^n$  be a "feasible" region,

The operator  $\mathcal{C}_{\mathbb{X}} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a **contractor** for  $\mathbb{X}$  if:

$$\forall \mathbf{x} \in \mathbb{R}^n, \begin{cases} \mathcal{C}_{\mathbb{X}}(\mathbf{x}) \subseteq \mathbf{x}, & \text{(contractance)} \\ \mathcal{C}_{\mathbb{X}}(\mathbf{x}) \cap \mathbb{X} = \mathbf{x} \cap \mathbb{X}. & \text{(completeness)} \end{cases}$$

# Definition: Contractor

Let  $\mathbb{X} \subseteq \mathbb{R}^n$  be a "feasible" region,

The operator  $\mathcal{C}_{\mathbb{X}} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a **contractor** for  $\mathbb{X}$  if:

$$\forall \mathbf{x} \in \mathbb{R}^n, \begin{cases} \mathcal{C}_{\mathbb{X}}(\mathbf{x}) \subseteq \mathbf{x}, & \text{(contractance)} \\ \mathcal{C}_{\mathbb{X}}(\mathbf{x}) \cap \mathbb{X} = \mathbf{x} \cap \mathbb{X}. & \text{(completeness)} \end{cases}$$

## Example:

The operator  $\mathcal{C} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a contractor for the equation  $f(x) = 0$ , if:

$$\forall \mathbf{x} \in \mathbb{R}^n, \begin{cases} \mathcal{C}(\mathbf{x}) \subseteq \mathbf{x}, \\ x \in \mathbf{x} \text{ and } f(x) = 0 \Rightarrow x \in \mathcal{C}(\mathbf{x}). \end{cases}$$

# Algebra on Contractors

Let  $\mathcal{A}$  a contractor for the equation  $f(x) = 0$ , and  $\mathcal{B}$  a contractor for the equation  $g(x) = 0$ , then:

## Intersection, Composition

$\mathcal{A} \cap \mathcal{B}$  and  $\mathcal{A} \circ \mathcal{B}$  are two contractors for the region:

$$\{x \in \mathbb{R}^n : f(x) = 0 \text{ AND } g(x) = 0\}$$

## Union

$\mathcal{A} \cup \mathcal{B}$  is a contractor for the region:

$$\{x \in \mathbb{R}^n : f(x) = 0 \text{ OR } g(x) = 0\}$$

# Algebra on Contractors

Let  $\mathcal{A}$  a contractor for the equation  $f(x) = 0$ , and  $\mathcal{B}$  a contractor for the equation  $g(x) = 0$ , then:

## Intersection, Composition

$\mathcal{A} \cap \mathcal{B}$  and  $\mathcal{A} \circ \mathcal{B}$  are two contractors for the region:

$$\{x \in \mathbb{R}^n : f(x) = 0 \text{ AND } g(x) = 0\}$$

## Union

$\mathcal{A} \cup \mathcal{B}$  is a contractor for the region:

$$\{x \in \mathbb{R}^n : f(x) = 0 \text{ OR } g(x) = 0\}$$

# Implementation

- Arithmetic:
  - **Interval Arithmetic:** 4 versions  
Filib++, Gaol, Profil, 1 homemade.
  - **Affine Arithmetics:** 7 versions  
fast/reliable, dynamic/static, Floating point/Interval
- Contractor:
  - Forward-Backward Contractor,
  - Convex Hull Contractor based on Linear Relaxation,
  - Contractor with quantifier,
  - "Non-Mathematical" Contractor,
  - ... your own contractor.
- Python3 Interface and userfriendly Installation:

```
pip install pyIbex
```

# Contents

## ① Concepts

Set-membership approach

Contractor

Properties

Implementation

## ② IBEXopt

Constraint Satisfaction Problem

Global Optimisation

## ③ Related Project

DynIBEX

ViabIBEX

$H_\infty$  control synthesis

# Example 1/3

Let us consider the following equations:

$$S_1 = \left\{ (x, y) \in \mathbb{R}^2 \mid \left( \left( \frac{x}{4} \right)^2 + \left( \frac{y}{3} \right)^2 - 1 \leq 0 \right) \text{ AND } (y \geq 0) \right\}$$

$$S_2 = \left\{ (x, y) \in \mathbb{R}^2 \mid \left( x - \frac{y}{2} \geq -4 \right) \text{ AND } \left( x + \frac{y}{2} \leq 4 \right) \text{ AND } (y \in [-4, 0]) \right\}$$

$$S_3 = \left\{ (x, y) \in \mathbb{R}^2 \mid \left( \left( \frac{x+1.8}{3.5} \right)^2 + \left( \frac{y+0.3}{2} \right)^2 \geq 0.1 \right) \text{ AND } \left( \left( \frac{x-1.8}{3.5} \right)^2 + \left( \frac{y+0.3}{2} \right)^2 \geq 0.1 \right) \text{ AND } (x^2 \right.$$

$$S_4 = \left\{ (x, y) \in \mathbb{R}^2 \mid (x - y \leq 1) \text{ OR } (x + y \geq -1) \text{ OR } (y \leq -2.8) \text{ OR } (\cos(10x) \in [0, 2]) \right\}$$

$$S_5 = \left\{ (x, y) \in \mathbb{R}^2 \mid \left( (x + y \geq -4) \text{ AND } (x - y \leq 4) \right) \text{ OR } \left( y + \frac{\cos 1.1x}{2} \geq -3 \right) \right\}$$

$$S_6 = \left\{ (x, y) \in \mathbb{R}^2 \mid \left( \left( \frac{5x}{3} + y - 6 \right) (y - \frac{x}{3}) \notin [0.4999, 0.5001] \right) \text{ AND } (y \leq 1) \text{ AND } (x \geq 0) \right\}$$

$$S_7 = \left\{ (x, y) \in \mathbb{R}^2 \mid \left( \left( -\frac{5x}{3} + y - 6 \right) (y + \frac{x}{3}) \notin [0.4999, 0.5001] \right) \text{ AND } (y \leq 1) \text{ AND } (x \leq 0) \right\}$$

$$S_8 = \left\{ (x, y) \in \mathbb{R}^2 \mid y \geq 1 \right\}$$

$$S_9 = \left\{ (x, y) \in \mathbb{R}^2 \mid (|x| \notin [0.1999, 0.2001]) \text{ AND } (y \geq -0.1) \right\}$$

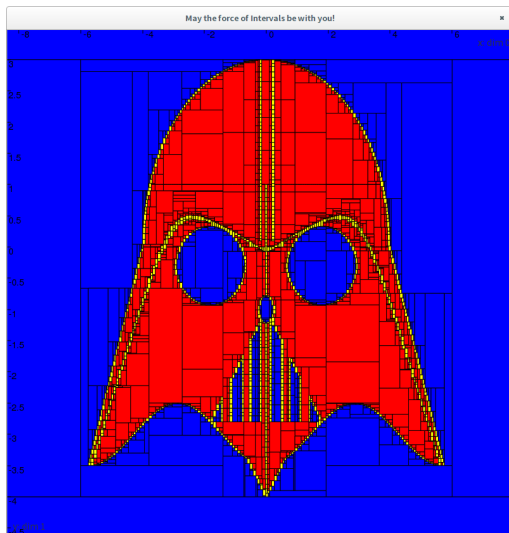
$$S_{10} = \left\{ (x, y) \in \mathbb{R}^2 \mid y \leq 0.1 \right\}$$

Draw the area defined by the following set  $S$ .

$$S = (S_1 \cup S_2) \cap S_3 \cap S_4 \cap S_5 \cap (S_6 \cup S_7 \cup S_8) \cap (S_9 \cup S_{10})$$

## Constraint Satisfaction Problem

## Example 2/3





## Example 3/3

```

sep1=SepFwdBwd(f("(x/4)^2+(y/3)^2-1"), Interval(-∞,0)) & SepFwdBwd
sep2=SepFwdBwd(f("x-y/2"), Interval(-4,∞)) & SepFwdBwd(f("x+y/2")),
sep3=SepFwdBwd(f("((x+1.8)/3.5)^2+((y+0.3)/2)^2"), Interval(0.1,∞))
sep4=SepFwdBwd(f("x-y"), Interval(-∞,1)) | SepFwdBwd(f("x+y"), Inte
sep5=SepFwdBwd(f("x+y"), Interval(-4,∞)) & SepFwdBwd(f("x-y"), Inte
SepFwdBwd(f("y+cos(1.1*x)/2"), Interval(-3,∞))
sep6=SepNot(SepFwdBwd(f("(5*x/3+y-6)*(y-x/3)"), Interval(0.4999,0.5
SepFwdBwd(f("x"), Interval(0,∞))
sep7=SepNot(SepFwdBwd(f("(-5*x/3+y-6)*(y+x/3)"), Interval(0.4999,0.
sep8=SepFwdBwd(f("y"), Interval(1,∞))
sep9=SepNot(SepFwdBwd(f("abs(x)"), Interval(0.1999,0.2001))) & SepF
sep10=SepFwdBwd(f("y"), Interval(-∞,0.1))

sep=(sep1|sep2) & sep3 & sep4 & sep5 & (sep6|sep7|sep8) & (sep9|se

pySIVIA(IntervalVector([[ -10, 10], [ -10, 10]]), sep, epsilon)

```

# Global Optimization

We consider global optimization of Non Linear Programming problems in a deterministic and reliable way.

## Problem

$$\begin{cases} \min_{x \in X \subset \mathbb{R}^n} & f(x) \\ \text{s.t.} & g_l(x) \leq 0, \forall l \in \{1, \dots, p\}, \\ & h_k(x) = 0, \forall k \in \{1, \dots, q\}. \end{cases}$$

## Modelization

- *AMPL*
- formal tool of *IBEX*

# Branch and Bound Algorithm

Each iteration:

- **Choice** and **Subdivision of the box  $X$**  (into 2 boxes),  
 $\implies \mathcal{L}$  list of possible solutions
- **Computation of lower bounds**  
 $\implies$  Interval Arithmetic, Affine Arithmetic,...
- **Elimination** of boxes that cannot contain the global optimum  
 $\implies$  Elts which do not satisfy constraints, lower bound  $> \tilde{f}$ ,..  
 Else: **Store in  $\mathcal{L}$**
- **STOP**  $\implies \max_{(\mathbb{Z}, f_z) \in \mathcal{L}} \text{wid}(\mathbb{Z}) \leq \epsilon_L$   
 $\implies \tilde{f} - \min_{(\mathbb{Z}, f_z) \in \mathcal{L}} f_z \leq \epsilon_f$

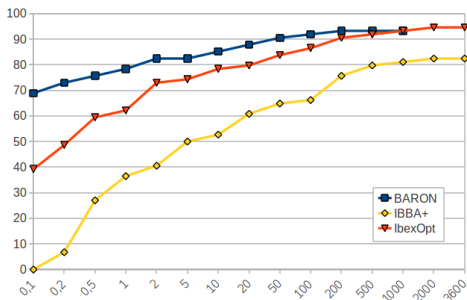
# Branch and Bound Algorithm

Each iteration:

- Choice and Subdivision of the box  $\mathbf{X}$  (into 2 boxes),  
 $\implies \mathcal{L}$  list of possible solutions
- Contract each sub-boxes,  
 $\implies \text{CtcAcid} \cap (\text{CtcPolytopeHull} \cap \text{CtcHC4})^\infty$ .
- Computation of lower bounds  
 $\implies$  Interval Arithmetic, Affine Arithmetic,...
- Elimination of boxes that cannot contain the global optimum  
 $\implies$  Elts which do not satisfy constraints, lower bound  $> \tilde{f}$ ,..  
 Else: Store in  $\mathcal{L}$
- STOP  $\implies \max_{(\mathbb{Z}, f_z) \in \mathcal{L}} \text{wid}(\mathbb{Z}) \leq \epsilon_L$   
 $\implies \tilde{f} - \min_{(\mathbb{Z}, f_z) \in \mathcal{L}} f_z \leq \epsilon_f$

# Performance Profile: BARON, IBEXopt, IBBA

Higher is better



⇒ Still need progress **BUT** we can deal with a more large variety of problems with cos, sin, atan,...

# Contents

## ① Concepts

Set-membership approach

Contractor

Properties

Implementation

## ② IBEXopt

Constraint Satisfaction Problem

Global Optimisation

## ③ Related Project

DynIBEX

ViabIBEX

$H_\infty$  control synthesis

# DynIBEX: A. Chapoutot and J. Alexandre dit Sandretto

## Guaranteed numerical integration method for ODE:

Ordinary Differential Equation with a given initial condition

We consider an initial value problem (IVP):

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \quad \text{with} \quad \mathbf{y}(0) = \mathbf{y}_0$$

The goal is to compute the sequence of boxes  $(t_n, [\mathbf{y}_n])_{n \in \mathbb{N}}$  such that  $[\mathbf{y}_n] \supseteq \mathbf{y}(t_n; [\mathbf{y}_{n-1}]) = \{\mathbf{y}(t_n; \mathbf{y}_{n-1}) : \forall \mathbf{y}_{n-1} \in [\mathbf{y}_{n-1}]\}$ .

Differential-algebraic equations in Hessenberg index 1 form with consistent initial conditions

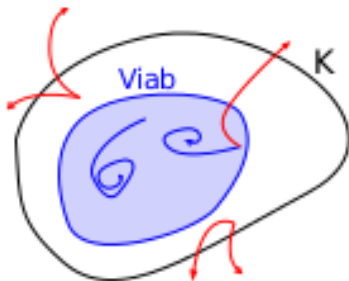
$$\begin{aligned} \dot{\mathbf{y}}(t) &= f(t, \mathbf{y}(t), \mathbf{x}(t)) \\ 0 &= g(t, \mathbf{y}(t), \mathbf{x}(t)) \end{aligned} \quad \text{with} \quad \mathbf{y}(0) = \mathbf{y}_0 \quad \text{and} \quad \mathbf{x}(0) = \mathbf{x}_0$$

# ViabIBEX: D. Monnet, L. Jaulin, J. Ninin

System  $\mathcal{S}$  defined by:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$$

A state  $\mathbf{x}$  is viable if at least one evolution of  $\mathcal{S}$  from  $\mathbf{x}$  can stay indefinitely in a set of constraints  $\mathbb{K}$ .





# ViabiBEX: D. Monnet, L. Jaulin, J. Ninin

MainWindow

Show Vector Field 0

More options

V-viability Polygon Integration

Fast computation

Nb control 3

Ignore viable points

Evolution Function

$$x^2 - 9.81 \cdot \sin\left(\frac{(1.1 \cdot \sin(1.2 \cdot x_1) - 1.2 \cdot \sin(1.1 \cdot x_1))}{2}\right) - 0.7 \cdot x^2 + u$$

Inner Constraints / Target

Outer Constraints / Dead Zone

Ignore equilibrium point4\4: (4.29513,0), in viable set

Clear log

7 viable sets found in 0.000569 seconds  
25.1205 % of space viable, 0 % of space not viable  
25.1205 % of space characterized

# ViabiBEX: D. Monnet, L. Jaulin, J. Ninin

The screenshot shows the ViabiBEX software interface. The main window displays a 2D plot of a viable set (green) and non-viable set (blue) on a grid. The viable set is a complex, irregular shape. The non-viable set is the complement of the viable set. The interface includes several control panels and a log window.

**Left Panel (Control Buttons):**

- V-viable Functions
- Apply Constraints
- Polygon Expansion
- Inner Approx** (highlighted with a red box)
- Outer Approx
- Stop

**Top Panel:**

- Show Vector Field (checkbox)
- 0 (slider)
- More options (button)

**Right Panel (Integration Settings):**

- V-viability | Polygon | Integration (tabs)
- Simulation time: 0.435815
- Simulation precision: 0.001
- Guaranteed integration
- Simulation method: HEUN (dropdown)

**Evolution Function:**

$$x_2 - 9.81 \cdot \sin\left(\frac{1.1 \cdot \sin(1.2 \cdot x_1) - 1.2 \cdot \sin(1.1 \cdot x_1)}{2}\right) - 0.7 \cdot x_2 + u$$

**Inner Constraints / Target:**

**Outer Constraints / Dead Zone:**

**Log Window:**

Ignore equilibrium point4\4: (4.29513,0), in viable set

Clear log

7 viable sets found in 0 seconds  
25.1205 % of space viable, 0 % of space not viable  
25.1205 % of space characterized

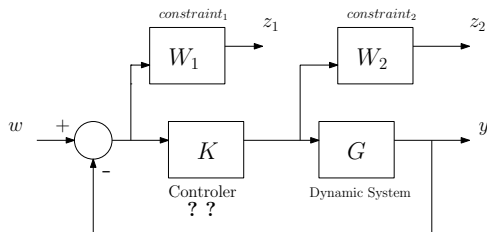
# ViabiBEX: D. Monnet, L. Jaulin, J. Ninin

The screenshot displays the ViabiBEX application window. The main area is a 2D plot with a grid. A large green region represents the viable set, bounded by a curved line. The rest of the plot is colored red or blue. The interface is annotated with red boxes and numbers 1 through 6:

- 1**: Points to the green viable set in the plot.
- 2**: Points to the status window at the bottom, which contains the following text:
 

```
Cannot solve  $A'W+WA = -I$ , try again placing pole at -1
suced to place poles at -1
failed: Cannot prove vector entrance
80 viable sets found in 127.685 seconds
4.79431 % of space viable, 0 % of space not viable
4.79431 % of space characterized
```
- 3**: Points to the right-hand control panel, which includes input fields for:
  - x1 min: -10, x1 max: 10
  - x2 min: -10, x2 max: 10
  - umin: -1, umax: 1
  - eps: 0.1
  - Evolution Function: x2, u
  - Inner Constraints / Target: (empty)
  - Outer Constraints / Dead Zone: (empty)
- 4**: Points to the "Show Vector Field" slider, which is set to 1.
- 5**: Points to the "Stop" button in the left-hand menu.
- 6**: Points to the "More options" checkbox, which is currently unchecked.

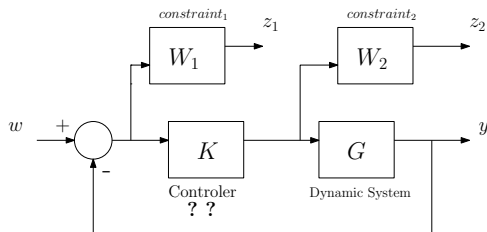
## SynthIBEX: D. Monnet, J. Ninin, C. Clément



$H_\infty$  control synthesis  $\Rightarrow$  Guarantee the robustness and stability

$$\|P\|_\infty = \sup_{\omega} (\sigma_{\max}(P(j\omega)))$$

## SynthIBEX: D. Monnet, J. Ninin, C. Clément



$H_\infty$  control synthesis  $\Rightarrow$  Guarantee the robustness and stability

$$\|P\|_\infty = \sup_{\omega} (\sigma_{\max}(P(j\omega)))$$

- Classical approach without structural constraint (convex problem)  $\Rightarrow$  LMI system, SDP optimization
- Classical approach **with structural constraint** (non-convex problem)  $\Rightarrow$  Nonsmooth **local** optimization [Apkarian and Noll, 2006]

# SynthIBEX: D. Monnet, J. Ninin, C. Clément

$$\left\{ \begin{array}{l} \min_{\mathbf{k}} \sup_{\omega} \max \left( \left\| \frac{W_1(j\omega)}{1 + G(j\omega)K(j\omega)} \right\|_{\infty}, \left\| \frac{W_2(j\omega)K(j\omega)}{1 + G(j\omega)K(j\omega)} \right\|_{\infty} \right) \\ \\ \text{The closed-loop system must be } \textit{stable}. \end{array} \right.$$

## Stability:

The system is stable iff its poles are strictly negative.

$\Leftrightarrow$

The roots of the denominator of  $\frac{1}{1+G(s)K(s)}$  are strictly negative

$\Leftrightarrow$

Routh-Hurwitz stability criterion

# SynthIBEX: D. Monnet, J. Ninin, C. Clément

$H_\infty$  control synthesis under structural constraint

$\Leftrightarrow$

Solve a **min/max problem with non-convex constraints**

$$\begin{cases} \min_{k \in \mathbb{K}} \sup_{\omega} f(k, \omega) \\ \text{s.t. } c_i(k) \leq 0, \forall i \in \{1, \dots, p\}, \end{cases}$$

## Branch and Bound algorithm for min/max problem

- Global optimization approach
- Guaranteed enclose of the global minimum
- Certificate of infeasibility

If you want to know more about Contractors

# IA MOOC

Interval Analysis MOOC  
with Luc JAULIN and Jordan NININ

<http://iamooc.ensta-bretagne.fr>

# IBEX

<http://www.ibex-lib.org>

**Fork it on GitHub**

<http://github.com/ibex-team/ibex-lib>

```
pip install pyIbex
```