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?

$\Longrightarrow$ Involve with the modelization improvement.

$\Longrightarrow$ 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?

$\Longrightarrow$ Involve with the modelization improvement.
$\Longrightarrow$ 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_∞ control synthesis

Contents

1. Concepts
   - Set-membership approach
   - Contractor
   - Properties
   - Implementation
2. IBEXopt
   - Constraint Satisfaction Problem
   - Global Optimisation
3. Related Project
   - DynIBEX
   - ViabIBEX
   - H_∞ control synthesis

Interval Arithmetic

Ideas

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

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

Interval Arithmetic

Ideas

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

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

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 $\mathcal{C}_{\mathbb{X}} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a contractor for $\mathbb{X}$ if:

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

Definition: Contractor

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

The operator $\mathcal{C}_{\mathbb{X}} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a contractor for $\mathbb{X}$ if:

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

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

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

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:

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

Union

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

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

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:

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

Union

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

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

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_∞ control synthesis

Example 1/3

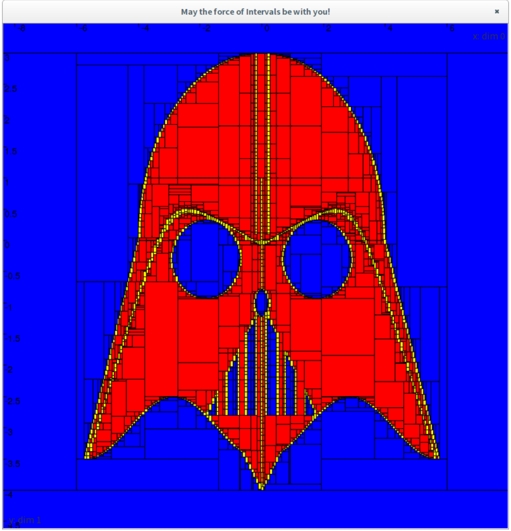
Let us consider the following equations:

$\mathbb{S}_1 = \left\{(x,y) \in \mathbb{R}^2 \mid \left(\left(\frac{x}{4}\right)^2 + \left(\frac{y}{3}\right)^2 - 1 \leqslant 0\right) \text{ AND } (y \geqslant 0)\right\}$
$\mathbb{S}_2 = \left\{(x,y) \in \mathbb{R}^2 \mid \left(x - \frac{y}{2} \geqslant -4\right) \text{ AND } \left(x + \frac{y}{2} \leqslant 4\right) \text{ AND } (y \in [-4, 0])\right\}$
$\mathbb{S}_3 = \left\{(x,y) \in \mathbb{R}^2 \mid \left(\left(\frac{x+1.8}{3.5}\right)^2 + \left(\frac{y+0.3}{2}\right)^2 \geqslant 0.1\right) \text{ AND } \left(\left(\frac{x-1.8}{3.5}\right)^2 + \left(\frac{y+0.3}{2}\right)^2 \geqslant 0.1\right) \text{ AND } (x^2$
$\mathbb{S}_4 = \left\{(x,y) \in \mathbb{R}^2 \mid (x - y \leqslant 1) \text{ OR } (x + y \geqslant -1) \text{ OR } (y \leqslant -2.8) \text{ OR } (\cos(10x) \in [0, 2])\right\}$
$\mathbb{S}_5 = \left\{(x,y) \in \mathbb{R}^2 \mid ((x + y \geqslant -4) \text{ AND } (x - y \leqslant 4)) \text{ OR } \left(y + \frac{\cos 1.1x}{2} \geqslant -3\right)\right\}$
$\mathbb{S}_6 = \left\{(x,y) \in \mathbb{R}^2 \mid \left(\left(\frac{5x}{3} + y - 6\right)\left(y - \frac{x}{3}\right) \notin [0.4999, 0.5001]\right) \text{ AND } (y \leqslant 1) \text{ AND } (x \geqslant 0)\right\}$
$\mathbb{S}_7 = \left\{(x,y) \in \mathbb{R}^2 \mid \left(\left(\frac{-5x}{3} + y - 6\right)\left(y + \frac{x}{3}\right) \notin [0.4999, 0.5001]\right) \text{ AND } (y \leqslant 1) \text{ AND } (x \leqslant 0)\right\}$
$\mathbb{S}_8 = \left\{(x,y) \in \mathbb{R}^2 \mid y \geqslant 1\right\}$
$\mathbb{S}_9 = \left\{(x,y) \in \mathbb{R}^2 \mid (|x| \notin [0.1999, 0.2001]) \text{ AND } (y \geqslant -0.1)\right\}$
$\mathbb{S}_{10} = \left\{(x,y) \in \mathbb{R}^2 \mid y \leqslant 0.1\right\}$

Draw the area defined by the following set $\mathbb{S}$.

$$\mathbb{S} = (\mathbb{S}_1 \cup \mathbb{S}_2) \cap \mathbb{S}_3 \cap \mathbb{S}_4 \cap \mathbb{S}_5 \cap (\mathbb{S}_6 \cup \mathbb{S}_7 \cup \mathbb{S}_8) \cap (\mathbb{S}_9 \cup \mathbb{S}_{10})$$

Example 2/3


May the force of Intervals be with you!
-8
-6
-4
-2
0
2
4
6
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4

Example 3/3

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

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

Modelization

- *AMPL*
- formal tool of *IBEX*

Branch and Bound Algorithm

Each iteration:

- Choice and Subdivision of the box **X** (into 2 boxes),
  $\Longrightarrow \mathcal{L}$ list of possible solutions

- Computation of lower bounds
  $\Longrightarrow$ Interval Arithmetic, Affine Arithmetic,...
- Elimination of boxes that cannot contain the global optimum
  $\Longrightarrow$ Elts which do not satisfy constraints, lower bound $> \widetilde{f}$,..
  Else: Store in $\mathcal{L}$
- STOP $\Longrightarrow \max\limits_{(\mathbb{Z},f_z)\in\mathcal{L}} wid(\mathbb{Z}) \leq \epsilon_L$
  $\Longrightarrow \widetilde{f} - \min\limits_{(\mathbb{Z},f_z)\in\mathcal{L}} f_z \leq \epsilon_f$

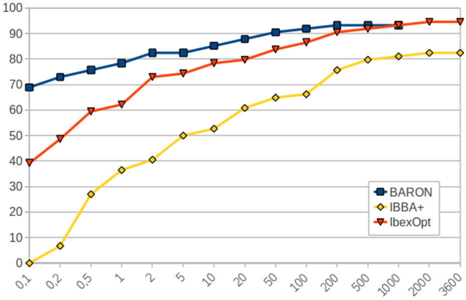
Branch and Bound Algorithm

Each iteration:

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

Performance Profile: BARON, IBEXopt, IBBA

Higher is better



$\Longrightarrow$ Still need progress BUT we can deal with a more large variaty 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_∞ 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

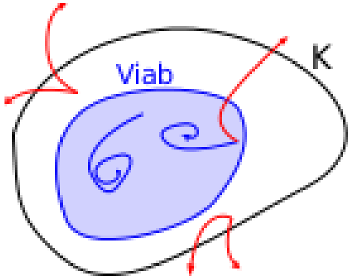
$$\begin{aligned} \dot{\mathbf{y}}(t) &= f(t, \mathbf{y}(t), \mathbf{x}(t)) \\ 0 &= g(t, \mathbf{y}(t), \mathbf{x}(t)) \end{aligned} \quad \text{with} \quad \mathbf{y}(0) = \mathbf{y}_0 \quad \text{and} \quad \mathbf{x}(0) = \mathbf{x}_0$$

ViablBEX: D. Monnet, L. Jaulin, J. Ninin

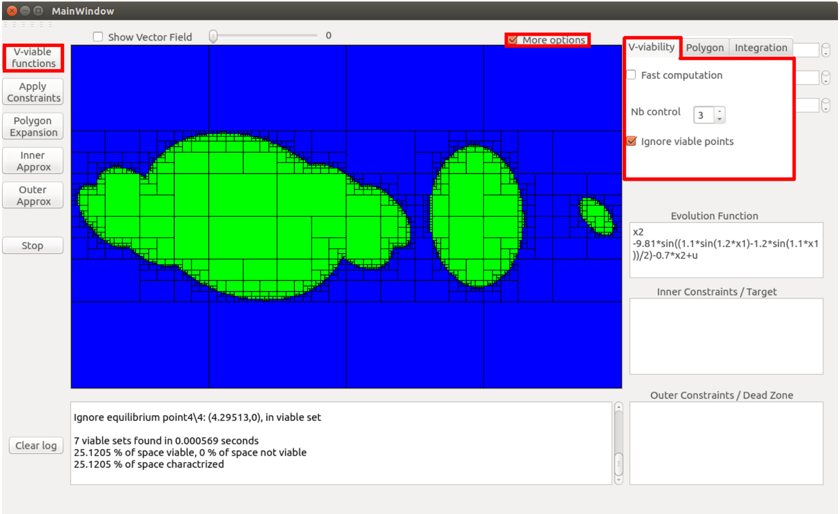
System $\mathcal{S}$ defined by:

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

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 $\mathbb{K}$.

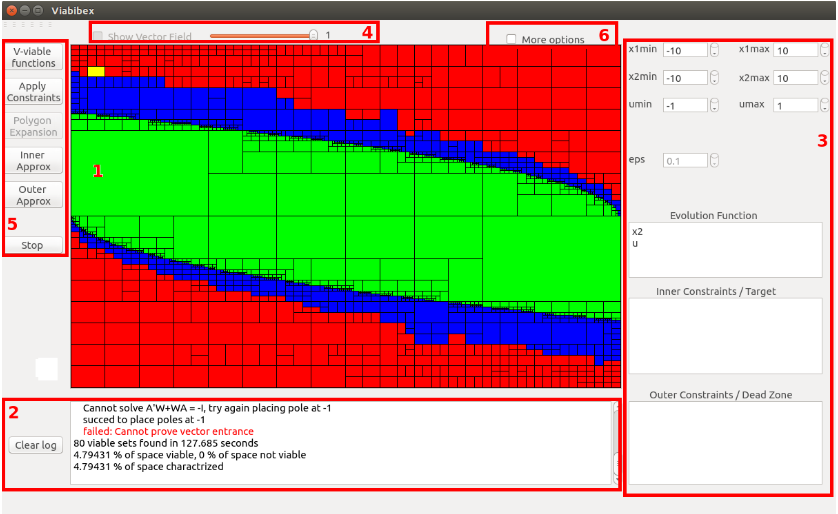


ViablIBEX: D. Monnet, L. Jaulin, J. Ninin

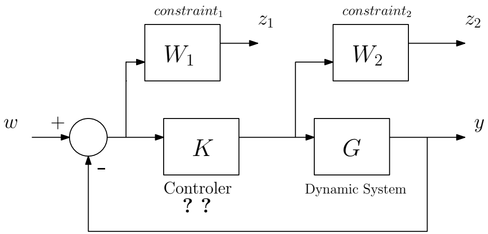


ViablBEX: D. Monnet, L. Jaulin, J. Ninin

ViablBEX: D. Monnet, L. Jaulin, J. Ninin



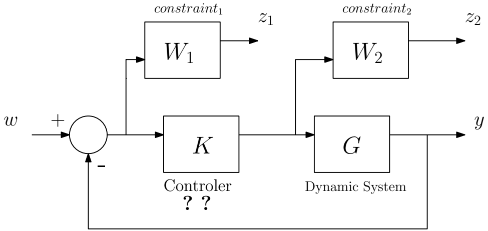
SynthIBEX: D. Monnet, J. Ninin, C. Clément



H_∞ 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_∞ control synthesis $\Rightarrow$ Guarantee the robustness and stability

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

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

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

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

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_∞ control synthesis under structural constraint

$$\Leftrightarrow$$

Solve a min/max problem with non-convex constraints

$$\begin{cases} \min\limits_{k \in \mathbb{K}} & \sup\limits_{\omega} f(k,\omega) \\ \text{s.t.} & c_i(k) \leq 0 \ , \ \forall i \in \{1, ..., p\}, \end{cases}$$

Branch and Bound algorithm for min/max problem

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

If you want to know more about Contractors

IAMOOC

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
```