Interval analysis with application to robust control of linear systems

Winter school, Lille, February 13, 2024

L. Jaulin

Interval analysis

Problem. Given $f : \mathbb{R}^n \to \mathbb{R}$ and a box $[\mathbf{x}] \subset \mathbb{R}^n$, prove that

$$\forall \mathbf{x} \in [\mathbf{x}], f(\mathbf{x}) \geq 0.$$

Interval arithmetic can solve efficiently this problem.

Interval arithmetic

$$
\begin{aligned}
[-1,3]+[2,5] &= ?,\\
[-1,3]\cdot[2,5] &= ?,
\end{aligned}
$$

Interval arithmetic

$$\begin{aligned} [-1,3]+[2,5] &= [1,8], \\ [-1,3]\cdot[2,5] &= [-5,15], \end{aligned}$$

Theorem (Moore, 1970)

$$[f]([\mathbf{x}]) \subset \mathbb{R}^{+} \Rightarrow \forall \mathbf{x} \in [\mathbf{x}], f(\mathbf{x}) \geq 0.$$

Interval arithmetic

If $\diamond \in \{+,-,\cdot,/,\max,\min\}$

$$[x]\diamond[y] \quad = \quad [\ \{x\diamond y \mid x\in[x], y\in[y]\}\ ].$$

where $[\mathbb{A}]$ is the smallest interval which encloses $\mathbb{A}\subset\mathbb{R}$.

If $f \in \{$cos, sin,sqr, sqrt, log, exp, $\dots\}$

$$f([x]) = [\{f(x) \mid x \in [x]\}].$$

Exercise.

$$\begin{array}{rcl} \sin([0,\pi]) & = & ? \\ \mathsf{sqr}([-1,3]) & = & [-1,3]^2 = ? \\ \mathsf{sqrt}([-10,4]) & = & \sqrt{[-10,4]} = ? \\ \log([-2,-1]) & = & ?. \end{array}$$

Solution.

$$\begin{array}{rcl} \sin([0,\pi]) & = & [0,1] \\ \mathsf{sqr}([-1,3]) & = & [-1,3]^2 = [0,9] \\ \mathsf{sqrt}([-10,4]) & = & \sqrt{[-10,4]} = [0,2] \\ \log([-2,-1]) & = & \emptyset. \end{array}$$

Inclusion functions

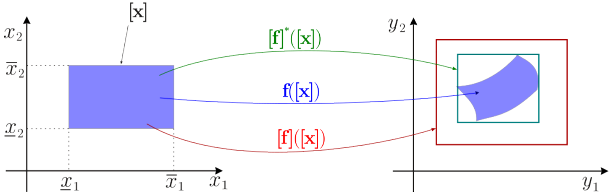
A *box*, or *interval vector* $[\mathbf{x}]$ of $\mathbb{R}^n$ is

$$[\mathbf{x}] = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+] = [x_1] \times \cdots \times [x_n].$$

The set of all boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$.

$[\mathbf{f}] : \mathbb{IR}^n \to \mathbb{IR}^m$ is an *inclusion function* for $\mathbf{f}$ if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n,\ \ \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]\,([\mathbf{x}]).$$



Inclusion functions $[\mathbf{f}]$ and $[\mathbf{f}]^*$

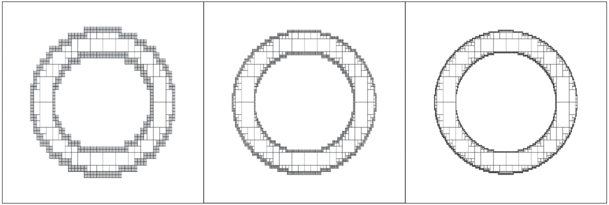
Set inversion

A subpaving of $\mathbb{R}^n$ is a set of non-overlapping boxes of $\mathbb{R}^n$.
Sets $\mathbb{X}$ can be bracketed between two subpavings:

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+.$$

Example.

$$\mathbb{X} = \{(x_1, x_2) \mid x_1^2 + x_2^2 \in [1, 2]\}.$$



Let $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^m$ and let $\mathbb{Y}$ be a subset of $\mathbb{R}^m$. Set inversion is the characterization of

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}).$$

We shall use the following tests.

$$\begin{array}{llll} \text{(i)} & [\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y} & \Rightarrow & [\mathbf{x}] \subset \mathbb{X} \\ \text{(ii)} & [\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset & \Rightarrow & [\mathbf{x}] \cap \mathbb{X} = \emptyset. \end{array}$$

Boxes for which these tests failed are bisected, except if they are too small.

Bode plot

Consider the linear system with delays:

$$\ddot{y}(t) - \ddot{y}(t-1) + 2\dot{y}(t) - \dot{y}(t-1) + y(t) = u(t)$$

In the Laplace domain, we have

$$s^2\hat{y}(s) - s^2e^{-s}\hat{y}(s) + 2s\hat{y}(s) - se^{-s}\hat{y}(s) + \hat{y}(s) = \hat{u}(s)$$

i.e.,

$$\left(s^2 - s^2e^{-s} + 2s - se^{-s} + 1\right)\hat{y}(s) = \hat{u}(s)$$

i.e.

$$\hat{y}(s) = \underbrace{\frac{1}{s^2 - s^2e^{-s} + 2s - se^{-s} + 1}}_{=H(s)} \cdot \hat{u}(s)$$

Its transfer function is

$$H(s) = \frac{1}{(s+1)(s(1-e^{-s})+1)}$$

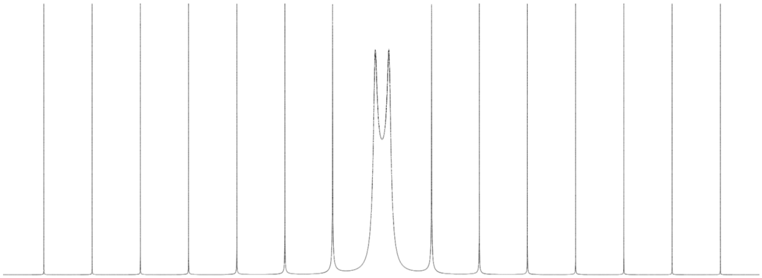
The gain is

$$\begin{aligned} h &= |H(j\omega)| \\ &\left|\frac{1}{(j\omega+1)} \cdot \frac{1}{(j\omega(1-e^{-j\omega})+1)}\right| \\ &= \left|\frac{1}{j\omega+1}\right| \cdot \left|\frac{1}{j\omega(1-\cos\omega+j\sin\omega)+1}\right| \\ &\frac{1}{\sqrt{1+\omega^2}} \cdot \frac{1}{\sqrt{(1-\omega\sin\omega)^2+\omega^2(1-\cos\omega)^2}} \end{aligned}$$

We ask SIVIA to characterize the set

$$\left\{(\omega,h)\in\mathbb{R}^2\,|\,h-|H(j\omega)|=0\right\}$$

```python
from codac import *
from vibes import vibes
f = Function("w","h",
             "h-1/((sqrt(1+w^2))*sqrt((1-w*sin(w))^2+w^2*(1-cos(w))^2))")
X=IntervalVector([[-50,50],[-1,3]])
S=SepFwdBwd(f,Interval(0))
vibes.beginDrawing()
SIVIA(X,S,0.004,color_map={SetValue.IN: "white[white]", SetValue.OUT: "white[white]",
                           SetValue.UNKNOWN: "black[black]"})
```



The spikes are almost impossible to get with classical plot methods

Stability domain

Consider the system

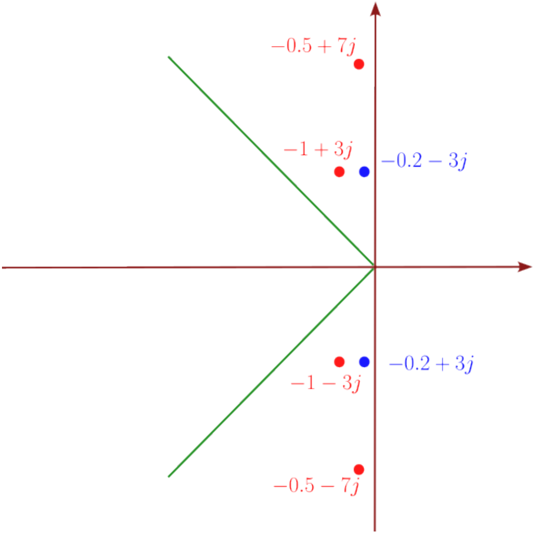
$$y^{(6)}+3.4y^{(5)}+71.49y^{(4)}+160.12\dddot{y}+1089.6\ddot{y}+1177.84\dot{y}+4452.2y=u$$

Its characteristic polynomial is

$$P(s)=s^6+3.4s^5+71.49s^4+160.12s^3+1089.6s^2+1177.84s+4452.2.$$

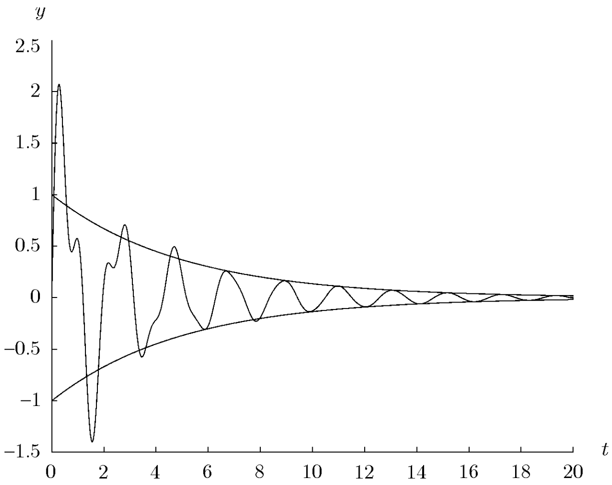
The roots are

$$\{-0.2-3j,-0.2+3j,-0.5-7j,-0.5+7j,-1-3j,-1+3j\}.$$


$-0.5+7j$
$-1+3j$
$-0.2-3j$
$-0.2+3j$
$-1-3j$
$-0.5-7j$

For $u = 0$, we have

$$\begin{aligned} y(t) &= \alpha_1 \sin(3t + \varphi_1) \exp(-0.2t) + \alpha_2 \sin(7t + \varphi_2) \exp(-0.5t) \\ &\quad + \alpha_3 \sin(3t + \varphi_2) \exp(-t) \end{aligned}$$



$\alpha_1 = \alpha_2 = \alpha_3 = 1$ and $\varphi_1 = \varphi_2 = \varphi_3 = 0$

The *stability domain* $\mathbb{S}_{\mathsf{p}}$ of the polynomial

$$P(s,\mathbf{p}) = s^n + a_{n-1}(\mathbf{p})s^{n-1} + \ldots + a_1(\mathbf{p})s + a_0(\mathbf{p})$$

is the set of all $\mathbf{p}$ such that $P(s,\mathbf{p})$ is stable.

Consider the example of Ackermann where

$$P(s,\mathbf{p}) = s^3 + (p_1+p_2+2)s^2 + (p_1+p_2+2)s + 2p_1p_2 + 6p_1 + 6p_2 + 2.25$$

Its Routh table is given by

1	p_1+p_2+2
p_1+p_2+2	$2p_1p_2+6p_1+6p_2+2.25$
$\frac{(p_1-1)^2+(p_2-1)^2-0.25}{p_1+p_2+2}$	0
$2(p_1+3)(p_2+3)-15.75$	0

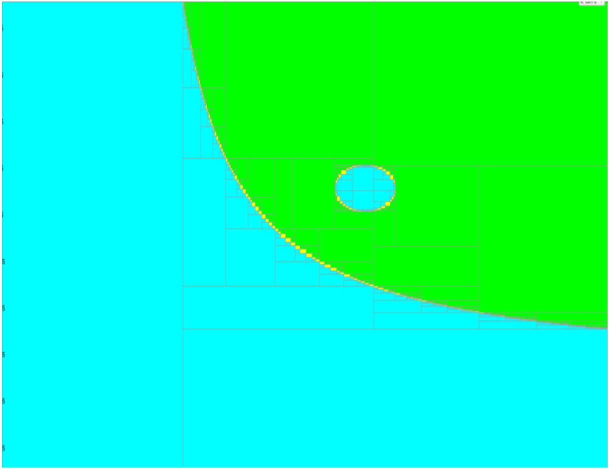
Its stability domain is thus defined by

$$\mathbb{S}_{\mathsf{p}} \triangleq \{\mathbf{p} \in \mathbb{R}^n \mid \mathbf{r}(\mathbf{p}) > \mathbf{0}\} = \mathbf{r}^{-1}\left(]0,+\infty[^{\times n}\right)$$

where

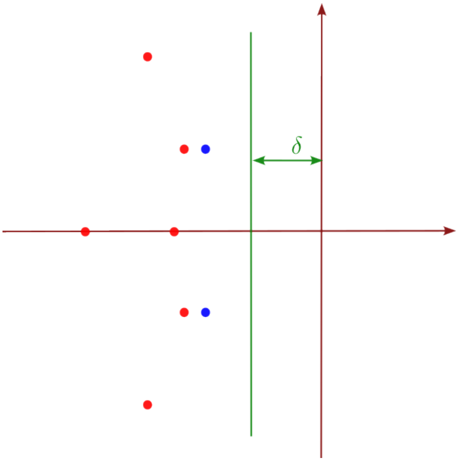
$$\mathbf{r}(\mathbf{p}) = \left(\begin{array}{c} p_1 + p_2 + 2 \\ (p_1 - 1)^2 + (p_2 - 1)^2 - 0.25 \\ 2(p_1 + 3)(p_2 + 3) - 15.75 \end{array} \right).$$

```
from codac import *
from vibes import vibes
f = Function("p1","p2",
             "(p1+p2+2;\
             (p1-1)^2+(p2-1)^2-0.25;\
             2*(p1+3)*(p2+3)-15.75)")
S=SepFwdBwd(f,IntervalVector(3,Interval(0,oo)))
P=IntervalVector([[-5,5],[-5,5]])
vibes.beginDrawing()
vibes.newFigure('Ackerman')
SIVIA(P,S,0.1)
```



Stability domain $\mathbb{S}_p$

Stability degrees



δ-stability

$P(s)$ is δ-stable if $P(s-\delta)$ is stable

Consider again the example of Ackermann where

$$P(s,\mathbf{p}) = s^3 + \underbrace{(p_1+p_2+2)}_{a_2}s^2 + \underbrace{(p_1+p_2+2)}_{a_1}s + \underbrace{2(p_1+3)(p_2+3)-13.75}_{a_0}$$

We have

$$\begin{array}{rl} & P(s-\delta,\mathbf{p}) \\ = & (s-\delta)^3 + a_2(s-\delta)^2 + a_1(s-\delta) + a_0 \\ = & s^3 - 3s^2\delta + 3\delta^2 s - \delta^3 + a_2(s^2 - 2s\delta + \delta^2) + a_1(s-\delta) + a_0 \\ = & s^3 + \underbrace{(a_2 - 3\delta)}_{b_2}s^2 + \underbrace{(3\delta^2 - 2a_2\delta + a_1)}_{b_1}s + \underbrace{-\delta^3 + a_2\delta^2 - a_1\delta + a_0}_{b_0} \end{array}$$

Its Routh table is given by

1	b_1
b_2	b_0
$b_1 - \frac{b_0}{b_2}$	0
b_0	0

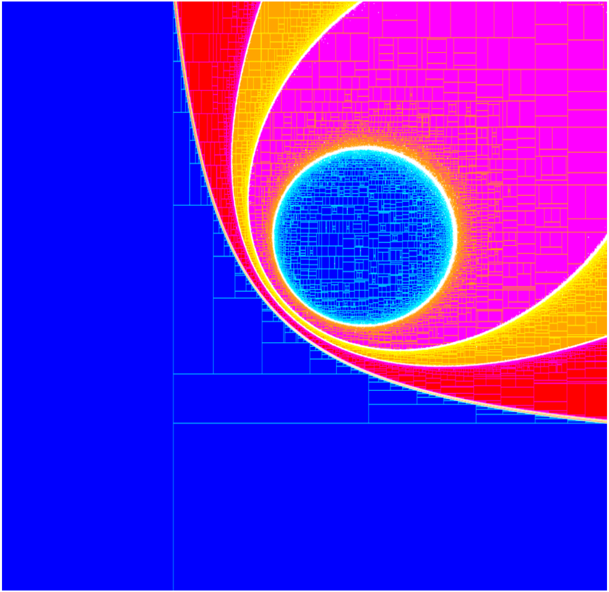
The δ stability domains are defined by

$$\mathbb{S}_{\mathbf{p}}(\delta) \triangleq \{\mathbf{p} \in \mathbb{R}^n \mid \mathbf{r}(\mathbf{p},\delta) > \mathbf{0}\}$$

where

$$\begin{aligned}
\mathbf{r}(\mathbf{p},\delta) &= \begin{pmatrix} b_2 \\ b_1 - \frac{b_0}{b_2} \\ b_0 \end{pmatrix} \\
\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} &= \begin{pmatrix} -\delta^3 + a_2\delta^2 - a_1\delta + a_0 \\ 3\delta^2 - 2a_2\delta + a_1 \\ a_2 - 3\delta \end{pmatrix} \\
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} &= \begin{pmatrix} 2(p_1+3)(p_2+3) - 13.75 \\ p_1 + p_2 + 2 \\ p_1 + p_2 + 2 \end{pmatrix}
\end{aligned}$$

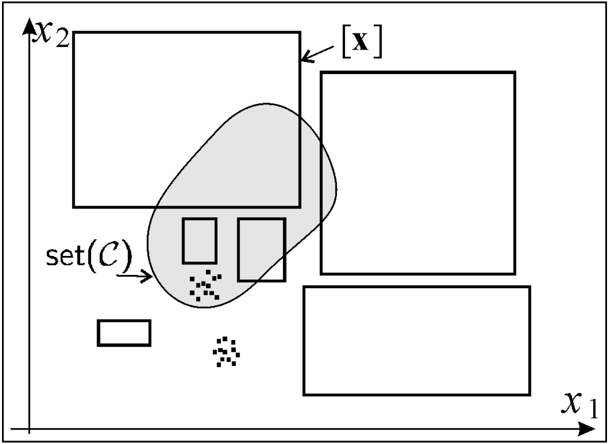
```
from codac import *
from vibes import vibes
vibes.beginDrawing()

R="(b2;b1-b0/b2;b0)"
R=R.replace("b2","(a2-3*d)");              R=R.replace("b1","(3*d^2-2*a2*d+a1)")
R=R.replace("b0","(-d^3+a2*d^2-a1*d+a0)"); R=R.replace("a2","(p1+p2+2)")
R=R.replace("a1","(p1+p2+2)");             R=R.replace("a0","(2*(p1+3)*(p2+3)-13.75)")
S0=SepFwdBwd(Function("p1","p2",R.replace("d","0")),  IntervalVector(3,Interval(0,oo)))
S1=SepFwdBwd(Function("p1","p2",R.replace("d","0.1")),IntervalVector(3,Interval(0,oo)))
S2=SepFwdBwd(Function("p1","p2",R.replace("d","0.2")),IntervalVector(3,Interval(0,oo)))

P=IntervalVector([[-5,5],[-5,5]])
SIVIA(P,S0,0.01,color_map={SetValue.IN: "orange[magenta]", SetValue.OUT: "cyan[blue]",
                 SetValue.UNKNOWN: "white[white]"})
SIVIA(P,S1,0.01,color_map={SetValue.IN: "yellow[orange]", SetValue.OUT: "transparent",
                 SetValue.UNKNOWN: "white[white]"})
SIVIA(P,S2,0.01,color_map={SetValue.IN: "magenta[red]", SetValue.OUT: "transparent",
                 SetValue.UNKNOWN: "white[white]"})
```

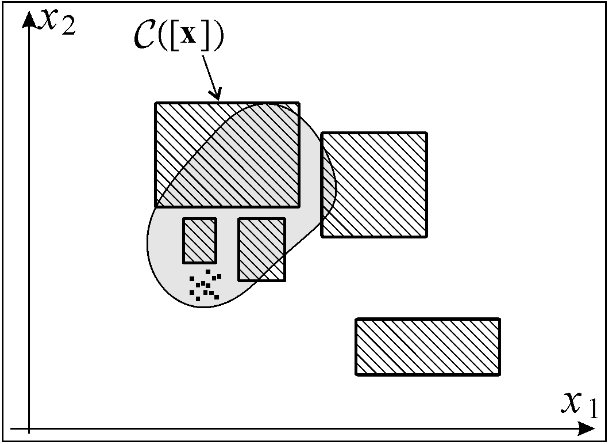


Contractors

The operator $\mathscr{C} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a *contractor* for $\mathbb{X} \subset \mathbb{R}^n$ if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathscr{C}([\mathbf{x}]) \subset [\mathbf{x}] & \text{(contractance),} \\ \mathscr{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} & \text{(completeness).} \end{cases}$$


x_2
[x]
set($\mathcal{C}$)
x_1


x_2
$\mathcal{C}([\mathbf{x}])$
x_1

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

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \begin{cases} \mathscr{C}([\mathbf{x}]) \subset [\mathbf{x}] \\ \mathbf{x} \in [\mathbf{x}] \text{ and } f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} \in \mathscr{C}([\mathbf{x}]) \end{cases}$$

Exercise. Let x, y, z be 3 variables such that

$$\begin{array}{rcl} x & \in & [-\infty, 5] \\ y & \in & [-\infty, 4] \\ z & \in & [6, \infty] \\ z & = & x + y. \end{array}$$

Contract the intervals for x, y, z.

Solution. We have

$$\begin{aligned} x &\in [x] = [2,5] \\ y &\in [y] = [1,4] \\ z &\in [z] = [6,9], \end{aligned}$$

Since $x \in [-\infty, 5], y \in [-\infty, 4], z \in [6, \infty]$ and $z = x + y$, we have

$$
\begin{array}{rrl}
z = x + y \Rightarrow & z \in & [6, \infty] \cap ([-\infty, 5] + [-\infty, 4]) \\
 & & = [6, \infty] \cap [-\infty, 9] = [6, 9]. \\
x = z - y \Rightarrow & x \in & [-\infty, 5] \cap ([6, \infty] - [-\infty, 4]) \\
 & & = [-\infty, 5] \cap [2, \infty] = [2, 5]. \\
y = z - x \Rightarrow & y \in & [-\infty, 4] \cap ([6, \infty] - [-\infty, 5]) \\
 & & = [-\infty, 4] \cap [1, \infty] = [1, 4].
\end{array}
$$

The contractor associated with $z = x + y$ is:

Algorithm pplus(inout: $[z],[x],[y]$)

$[z] := [z] \cap ([x] + [y])$ // $z = x + y$
$[x] := [x] \cap ([z] - [y])$ // $x = z - y$
$[y] := [y] \cap ([z] - [x])$ // $y = z - x$

The contractor associated with $z = x \cdot y$ is:

Algorithm pmult (inout: $[z],[x],[y]$)

$[z] := [z] \cap ([x] \cdot [y])$	// $z = x \cdot y$
$[x] := [x] \cap ([z] \cdot 1/[y])$	$//x = z/y$
$[y] := [y] \cap ([z] \cdot 1/[x])$	$//\ y = z/x$

The contractor associated with $y = \exp x$ is:

Algorithm pexp (inout: $[y], [x]$)

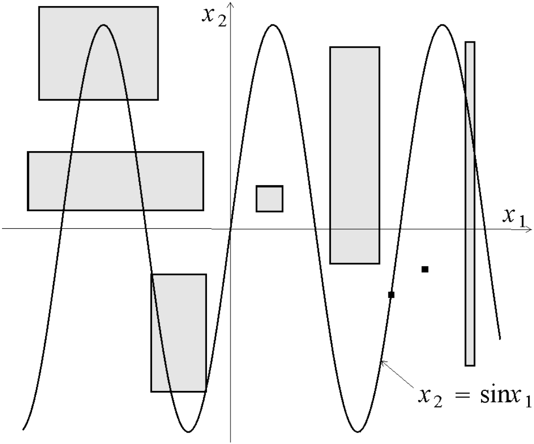
$[y] := [y] \cap \exp([x])$

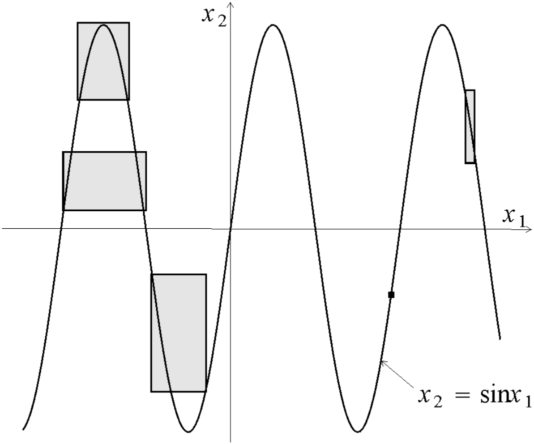
$[x] := [x] \cap \log([y])$

Any constraint for which such a projection procedure is available will be called a *primitive constraint*.

Example. Consider the primitive equation:

$$x_2 = \sin x_1.$$


x_2
x_1
$x_2 = \sin x_1$


x_2
x_1
$x_2 = \sin x_1$

Decomposition

$$x + \sin(xy) \leq 0,$$
$$x \in [-1, 1], y \in [-1, 1]$$

Decomposition

$$x + \sin(xy) \leq 0,$$
$$x \in [-1,1], y \in [-1,1]$$

can be decomposed into

$$\left\{ \begin{array}{l} a = xy \\ b = \sin(a) \\ c = x + b \end{array} \right. , \qquad \begin{array}{ll} x \in [-1,1] & a \in [-\infty,\infty] \\ y \in [-1,1] & b \in [-\infty,\infty] \\ & c \in [-\infty,0] \end{array}$$

Propagation

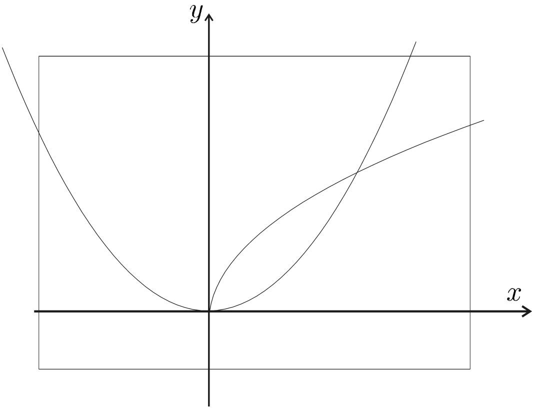
Consider the system of two equations.

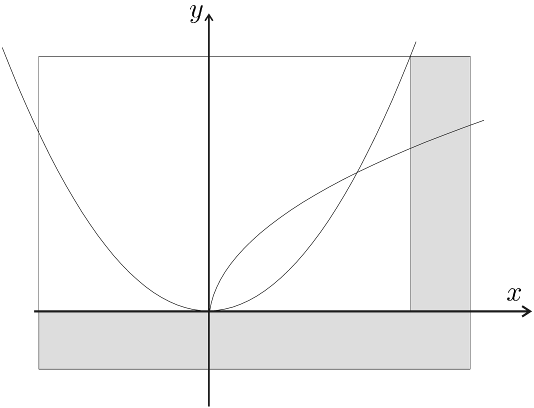
$$\begin{aligned} y &= x^2 \\ y &= \sqrt{x}. \end{aligned}$$

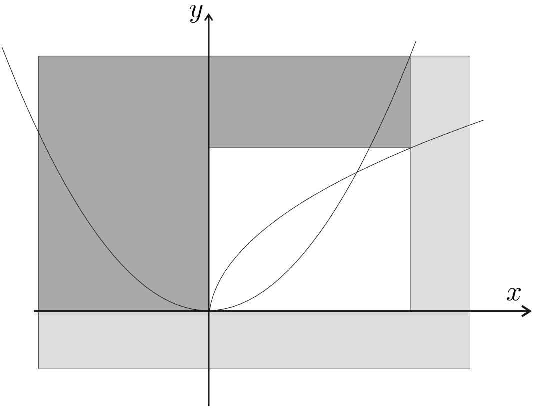
We can build two contractors

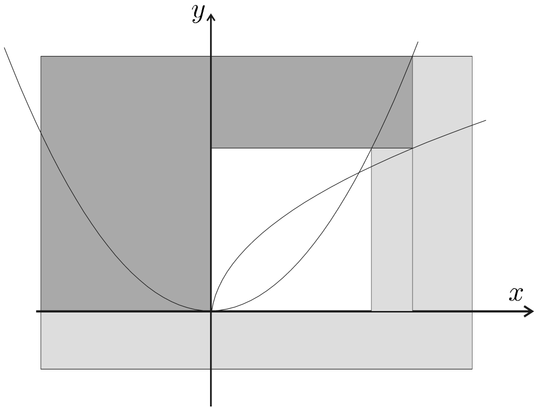
$$\mathscr{C}_1 : \left\{ \begin{array}{l} [y] = [y] \cap [x]^2 \\ [x] = [x] \cap \sqrt{[y]} \end{array} \right. \quad \text{associated to } y = x^2$$

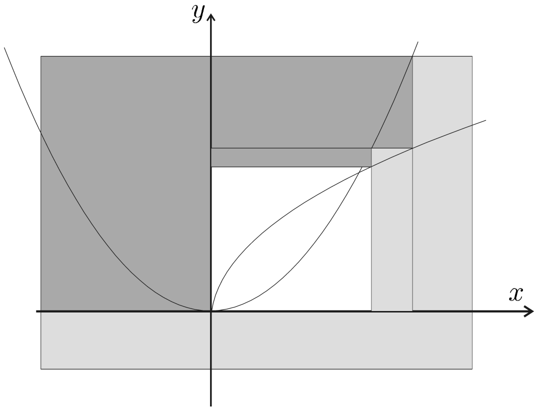
$$\mathscr{C}_2 : \left\{ \begin{array}{l} [y] = [y] \cap \sqrt{[x]} \\ [x] = [x] \cap [y]^2 \end{array} \right. \quad \text{associated to } y = \sqrt{x}$$

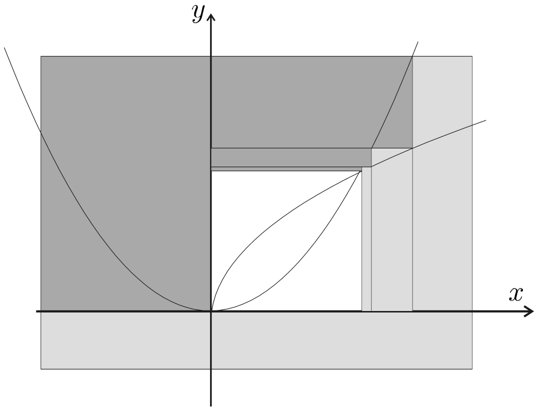

y
x

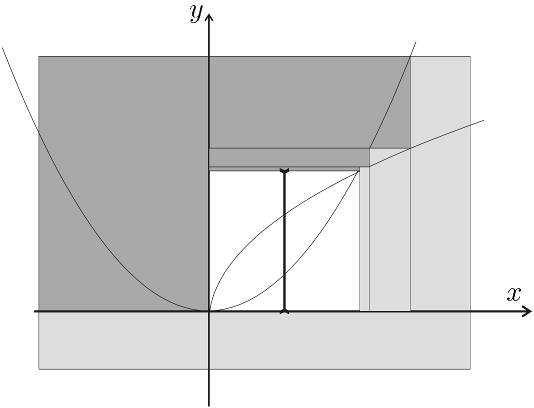

y
x

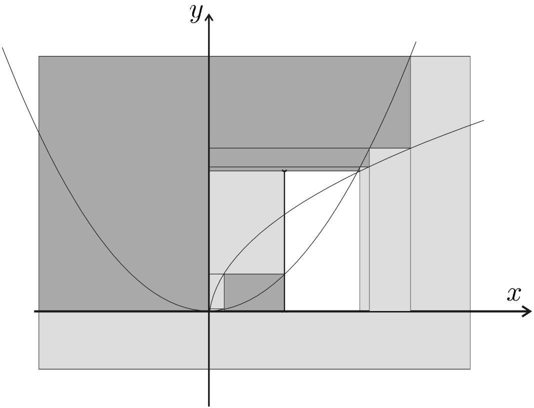

y
x

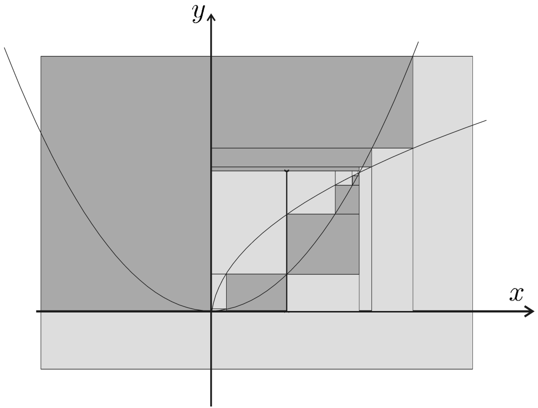

y
x


y
x


y
x

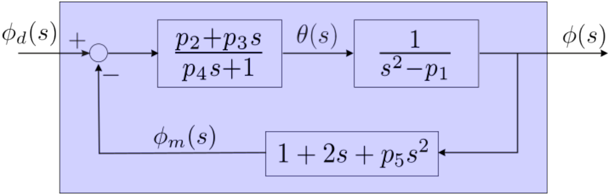

y
x


y
x


y
x

Robust stability

Consider a motorbike with a speed of 1m/s.
Angle of the handlebars: θ.
Rolling angle: ϕ
Wanted rolling angle: ϕ_d
Measured rolling angle: ϕ_m.



The input-output relation of the closed-loop system is :

$$\phi(s) = \frac{p_2 + p_3 s}{(s^2 - p_1)(p_4 s + 1) + (p_2 + p_3 s)(1 + 2s + p_5 s^2)} \cdot \phi_d(s).$$

Its characteristic polynomial is thus

$$\begin{aligned} P(s) &= \left(s^2 - p_1\right)(p_4 s + 1) + (p_2 + p_3 s)\left(1 + 2s + p_5 s^2\right) \\ &= a_3 s^3 + a_2 s^2 + a_1 s + a_0, \end{aligned}$$

with

$$\begin{array}{cc} a_3 = p_4 + p_3 p_5 & a_2 = p_2 p_5 + 2p_3 + 1 \\ a_1 = p_3 - p_1 p_4 + 2p_2 & a_0 = -p_1 + p_2. \end{array}$$

The Routh table is :

a_3	a_1
a_2	a_1
$\frac{a_2 a_1 - a_3 a_0}{a_2}$	0
a_0	0

The closed-loop system is stable if $a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}$ and a_0 have the same sign.

Assume that it is known that

$$p_1 \in [8.8; 9.2] \qquad p_2 \in [2.8; 3.2]$$
$$p_3 \in [0.8; 1.2] \qquad p_4 \in [1.8; 2.2]$$
$$p_5 \in [-3.2; -2.8].$$

The system is robustly stable if,

$\forall p_1 \in [p_1], \forall p_2 \in [p_2], \forall p_3 \in [p_3], \forall p_4 \in [p_4], \forall p_5 \in [p_5]$,
a_3, a_2, $\frac{a_2 a_1 - a_3 a_0}{a_2}$ and a_0 have the same sign.

Now, we have the equivalence

$$
\begin{array}{rl}
 & b_1,\ b_2,\ b_3 \text{ and } b_4 \text{ have the same sign} \\
\Leftrightarrow & \left\{ \begin{array}{ll} & \min(b_1,b_2,b_3,b_4) > 0 \\ \text{or} & \max(b_1,b_2,b_3,b_4) < 0 \end{array} \right.
\end{array}
$$

The robust stability condition amounts to proving that

$$\forall p_1 \in [p_1], \forall p_2 \in [p_2], \forall p_3 \in [p_3], \forall p_4 \in [p_4], \forall p_5 \in [p_5]$$
$$\begin{cases} & \min\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) > 0 \\ \text{or} & \max\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) < 0 \end{cases}$$

This means that the constraint network

$$\begin{array}{rcl} \mathscr{V} & = & \{p_1, p_2, p_3, p_4, p_5\}, \\ \mathscr{D} & = & \{[p_1], [p_2], [p_3], [p_4], [p_5]\}, \\ \mathscr{C} & = & \left\{ \begin{array}{l} a_3 = p_4 + p_3 p_5 \; ; \; a_2 = p_2 p_5 + 2 p_3 + 1 \\ a_1 = p_3 - p_1 p_4 + 2 p_2 \\ a_0 = -p_1 + p_2 \\ b_1 = \min\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) \\ b_2 = \max\left(a_3, a_2, \frac{a_2 a_1 - a_3 a_0}{a_2}, a_0\right) \\ b_1 \leq 0, \; b_2 \geq 0. \end{array} \right\} \end{array}$$

has no solution.

The proof of robust stability is done using codac

```
from codac import *
p1,p2,p3=Interval(8.8,9.2),Interval(2.8,3.2),Interval(0.8,1.2)
p4,p5=Interval(1.8,2.2),Interval(-3.2,-2.8)
b1,b2=Interval(-oo,0),Interval(0,oo)
a0,a1,a2,a3=Interval(-oo,oo),Interval(-oo,oo),Interval(-oo,oo),Interval(-oo,oo)
cn = ContractorNetwork()
cn.add(CtcFunction(Function("a3","p4","p3","p5", "p4+p3*p5-a3")),[a3,p3,p4,p5]);
cn.add(CtcFunction(Function("a2","p2","p3","p5", "p2*p5+2*p3+1-a2")),[a2,p2,p3,p5])
cn.add(CtcFunction(Function("a1","p4","p1","p2","p3", "p3-p1*p4+2*p2-a1")),[a1,p1,p2,p3,p4])
cn.add(CtcFunction(Function("a0","p1","p2","p2-p1-a0")),[a0,p1,p2])
cn.add(CtcFunction(Function("b1","a0","a1","a2","a3","min(a3,a2,(a2*a1-a3*a0)/a2,a0)-b1")),[b1,a0,a1,a2,a3])
cn.add(CtcFunction(Function("b2","a0","a1","a2","a3","max(a3,a2,(a2*a1-a3*a0)/a2,a0)-b2")),[b2,a0,a1,a2,a3])
cn.contract()
```

Separators

A *separator* $\mathscr{S}$ is pair of contractors $\{\mathscr{S}^{\text{in}}, \mathscr{S}^{\text{out}}\}$ such that

$$\mathscr{S}^{\text{in}}([\mathbf{x}]) \cup \mathscr{S}^{\text{out}}([\mathbf{x}]) = [\mathbf{x}] \qquad \text{(complementarity).}$$

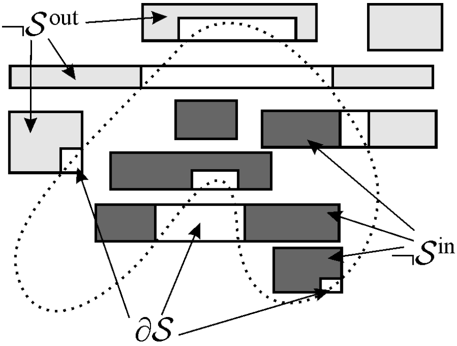
A set $\mathbb{S}$ is *consistent* with $\mathscr{S}$ (we write $\mathbb{S} \sim \mathscr{S}$), if

$$\mathbb{S} \sim \mathscr{S}^{\text{out}} \text{ and } \overline{\mathbb{S}} \sim \mathscr{S}^{\text{in}}.$$

The *remainder* of $\mathscr{S}$ is

$$\partial\mathscr{S}([\mathbf{x}]) = \mathscr{S}^{\mathsf{in}}([\mathbf{x}]) \cap \mathscr{S}^{\mathsf{out}}([\mathbf{x}]).$$

$\partial\mathscr{S}$ is a contractor, not a separator.



$\neg\mathscr{S}^{\text{in}}([\mathbf{x}])$, $\neg\mathscr{S}^{\text{out}}([\mathbf{x}])$ and $\partial\mathscr{S}([\mathbf{x}])$

Inclusion

$$\mathscr{S}_1 \subset \mathscr{S}_2 \Leftrightarrow \mathscr{S}_1^{\text{in}} \subset \mathscr{S}_2^{\text{in}} \text{ and } \mathscr{S}_1^{\text{out}} \subset \mathscr{S}_2^{\text{out}}.$$

Here $\subset$ means *more accurate.*

$\mathscr{S}$ is *minimal* if

$$\mathscr{S}_1 \subset \mathscr{S} \Rightarrow \mathscr{S}_1 = \mathscr{S}.$$

i.e., if $\mathscr{S}^{\text{in}}$ and $\mathscr{S}^{\text{out}}$ are both minimal.

Algebra

Contractor algebra only allows monotonic operations such as $\cup$ or $\cap$.

The complementary $\overline{\mathscr{C}}$ of a contractor $\mathscr{C}$, the restriction $\mathscr{C}_1 \backslash \mathscr{C}_2$, etc. cannot be defined.
Separators extend the operations allowed for contractors to non monotonic expressions.

The *complement* of $\mathscr{S} = \left\{\mathscr{S}^{\text{in}}, \mathscr{S}^{\text{out}}\right\}$ is

$$\overline{\mathscr{S}} = \left\{\mathscr{S}^{\text{out}}, \mathscr{S}^{\text{in}}\right\}.$$

If $\mathscr{S}_i = \left\{\mathscr{S}_i^{\text{in}}, \mathscr{S}_i^{\text{out}}\right\}, i \geq 1$, are separators, we define

$$\begin{array}{rclc}
\mathscr{S}_1 \cap \mathscr{S}_2 & = & \left\{\mathscr{S}_1^{\text{in}} \cup \mathscr{S}_2^{\text{in}}, \mathscr{S}_1^{\text{out}} \cap \mathscr{S}_2^{\text{out}}\right\} & \text{(intersection)} \\
\mathscr{S}_1 \cup \mathscr{S}_2 & = & \left\{\mathscr{S}_1^{\text{in}} \cap \mathscr{S}_2^{\text{in}}, \mathscr{S}_1^{\text{out}} \cup \mathscr{S}_2^{\text{out}}\right\} & \text{(union)} \\
\mathscr{S}_1 \backslash \mathscr{S}_2 & = & \mathscr{S}_1 \cap \overline{\mathscr{S}_2}. & \text{(difference)}
\end{array}$$

Theorem. If $\mathbb{S}_i$ are subsets of $\mathbb{R}^n$, we have

$$\begin{array}{ccc} \mathbb{S}_1 \cap \mathbb{S}_2 & \sim & \mathscr{S}_1 \cap \mathscr{S}_2 \\ \mathbb{S}_1 \cup \mathbb{S}_2 & \sim & \mathscr{S}_1 \cup \mathscr{S}_2 \\ \overline{\mathbb{S}_i} & \sim & \overline{\mathscr{S}_i} \\ \mathbb{S}_1 \backslash \mathbb{S}_2 & \sim & \mathscr{S}_1 \backslash \mathscr{S}_2 . \end{array}$$

Dealing with quantifiers

We can build a separator associated to the projection of a set defined by a separator.

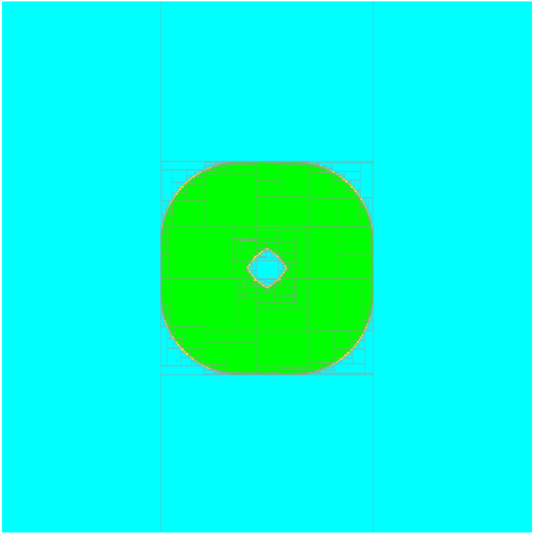
Oblong set. Consider the oblong set

$$\mathbb{X} = \left\{ \mathbf{x} \in \mathbb{R}^2 \,|\, \exists \mathbf{a} \in [-1,1]^2,\, (x_1 - a_1)^2 + (x_2 - a_2)^2 \in [4,9] \right\}$$

We build the separator $\mathscr{S}$ for $\mathbb{X}$ as follows

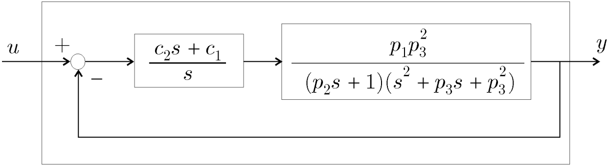
$$\begin{array}{lcl} \mathscr{S}_1([\mathbf{x}],[\mathbf{a}]) & \sim & (x_1 - a_1)^2 + (x_2 - a_2)^2 \in [4,9] \\ \mathscr{S}([\mathbf{x}]) & = & \underset{[\mathbf{a}]}{\text{proj}} \mathscr{S}_1([\mathbf{x}],[\mathbf{a}]) \end{array}$$

```python
from codac import *
from vibes import vibes
f = Function("x1","x2","a1","a2", "(x1-a1)^2+(x2-a2)^2")
S1=SepFwdBwd(f,Interval(4,9))
A=IntervalVector([[-1,1],[-1,1]])
S=SepProj(S1,A,0.01)
X=IntervalVector([[-10,10],[-10,10]])
vibes.beginDrawing()
SIVIA(X,S,0.1)
```



Robust control

Consider the system



with $\mathbf{p} \in [\mathbf{p}] = [0.9, 1.1]^{\times 3}$ and $\mathbf{c} \in [\mathbf{c}] = [0, 1]^2$.

$$\Sigma(\mathbf{p},\mathbf{c}) \text{ is stable } \Leftrightarrow r(\mathbf{c},\mathbf{p}) > 0.$$

Define

$$\mathbb{C} = \{\mathbf{c} \in [\mathbf{c}] \mid \forall \mathbf{p} \in [\mathbf{p}], r(\mathbf{c},\mathbf{p}) > 0\}$$

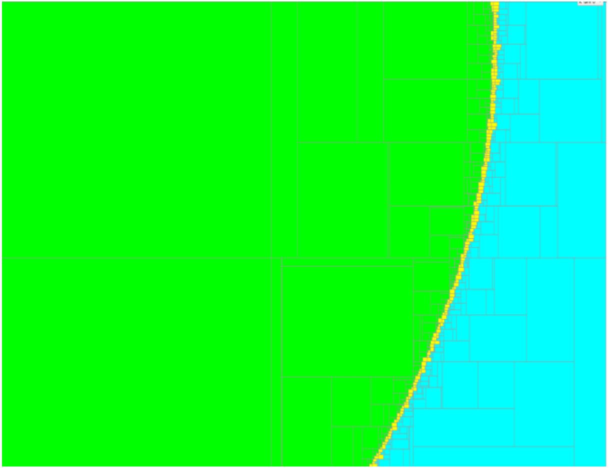
The transfer function of $\Sigma(\mathbf{p},\mathbf{c})$ is

$$H(s)=\frac{(c_2 s+c_1)\,p_1 p_3^2}{p_2 s^4+(p_2 p_3+1)\,s^3+\left(p_2 p_3^2+p_3\right)s^2+\left(p_3^2+c_2 p_1 p_3^2\right)s+c_1 p_1 p_3^2}$$

The first column of the corresponding Routh table is

$$\left(\begin{array}{c} p_2 \\ p_2 p_3+1 \\ p_2 p_3^2+p_3-\frac{p_2 p_3^2(1+c_2 p_1)}{p_2 p_3+1} \\ (1+c_2 p_1)\,p_3^2-\frac{(c_1 p_1 p_3)}{1-\frac{p_2 p_3(1+c_2 p_1)}{(p_2 p_3+1)^2}} \\ c_1 p_1 p_3^2 \end{array}\right)$$

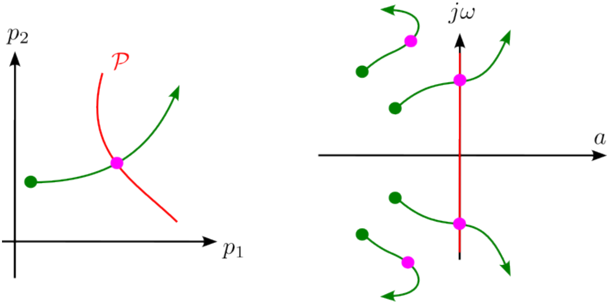
```
from codac import *
from vibes import vibes
f = Function("c1","c2","p1","p2","p3",
             "(p2; p2*p3+1;\
             p3*(p2*p3+1)^2-p2*p3^2*(1+c2*p1) ; \
             (1+c2*p1)*p3^2-(c1*p1*p3)/(1-p2*p3*(1+c2*p1)/((p2*p3+1)^2)) ;\
             c1*p1*p3^2)");
P=IntervalVector([[0.9,1.1],[0.9,1.1],[0.9,1.1]])
C0 =IntervalVector([[0,1],[0,1]])
S1=SepFwdBwd(f,IntervalVector(5,[0,oo]))
S=~SepProj(~S1,P)
vibes.beginDrawing()
SIVIA(C0,S,0.1)
```



Value set approach

The roots of $P(\mathbf{p},s)=0$ change continuously with $\mathbf{p}$.
We define the *value set*

$$\mathscr{P}=\{\mathbf{p}\,|\,\exists\omega>0,\,P(\mathbf{p},j\omega)=0\}.$$



Zero exclusion theorem

Cut off frequency. The roots of

$$P(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0$$

are in the disk with center 0 and radius

$$\omega_c = 1 + \max\{\|a_0\|, \|a_1\|, \ldots, \|a_{n-1}\|\}$$

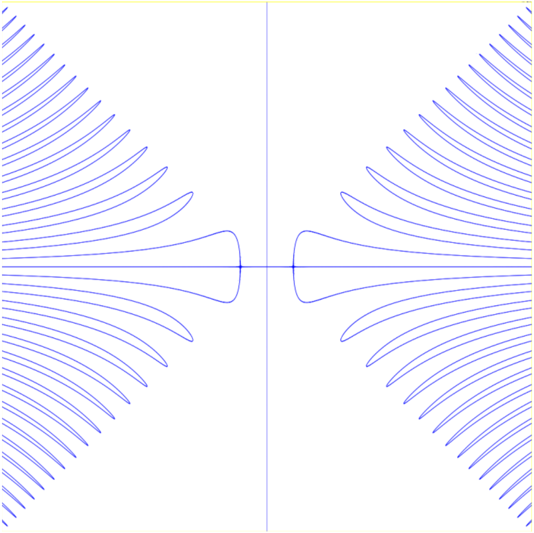
which is the Cauchy bound.

For

$$P(\mathbf{p},s) = s^3 + \sin(p_1p_2)\cdot s^2 + p_1^2\cdot s + p_1p_2$$

and $s = j\omega$, we get

$$\begin{array}{cc} & (j\omega)^3 + \sin(p_1p_2)\cdot(j\omega)^2 + p_1^2\cdot(j\omega) + p_1p_2 = 0 \\ \Leftrightarrow & -j\omega^3 - \sin(p_1p_2)\cdot\omega^2 + jp_1^2\cdot\omega + p_1p_2 = 0 \\ \Leftrightarrow & \left\{ \begin{array}{ccc} -\sin(p_1p_2)\cdot\omega^2 + p_1p_2 & = & 0 \\ -\omega^2 + p_1^2 & = & 0 \end{array} \right. \end{array}$$



Linear systems with delays

Turkulov system. Consider the system

$$\ddot{x}(t) + 2\dot{x}(t - p_1) + x(t - p_2) = 0$$

Its characteristic function is

$$P(\mathbf{p}, s) = s^2 + 2se^{-sp_1} + e^{-sp_2}.$$

We define

$$\mathscr{P} = \{\mathbf{p} \,|\, \exists \omega > 0, P(\mathbf{p}, j\omega) = 0\}.$$

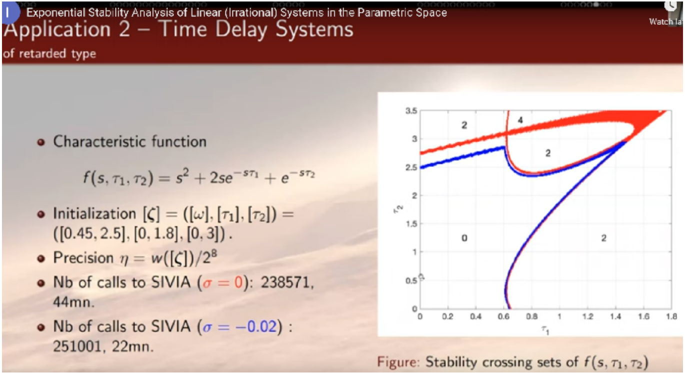
Now

$$\begin{array}{rl} & P(p_1,p_2,j\omega) \\ = & -\omega^2 + 2j\omega e^{-j\omega p_1} + e^{-j\omega p_2} \\ = & -\omega^2 + 2j\omega(\cos(\omega p_1) - j\sin(\omega p_1)) \\ & +\cos(\omega p_2) - j\sin(-\omega p_2) \\ = & -\omega^2 + 2\omega\sin(\omega p_1) + \cos(\omega p_2) \\ & +j\cdot(2\omega\cos(\omega p_1) - \sin(\omega p_2)) \end{array}$$

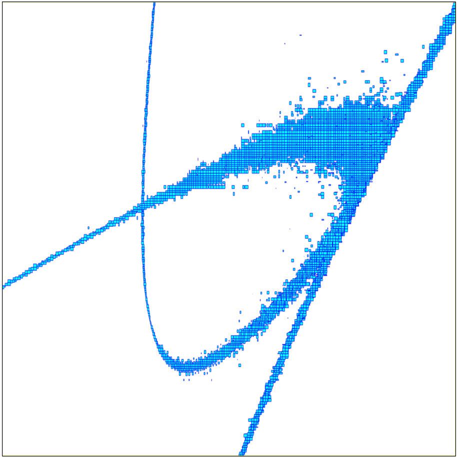
We have

$$\begin{aligned} & P(p_1,p_2,j\omega) = 0 \\ \Leftrightarrow \quad & \underbrace{\begin{pmatrix} -\omega^2 + 2\omega\sin(\omega p_1) + \cos(\omega p_2) \\ 2\omega\cos(\omega p_1) - \sin(\omega p_2) \end{pmatrix}}_{\mathbf{f}(p_1,p_2,\omega)} = \mathbf{0} \end{aligned}$$

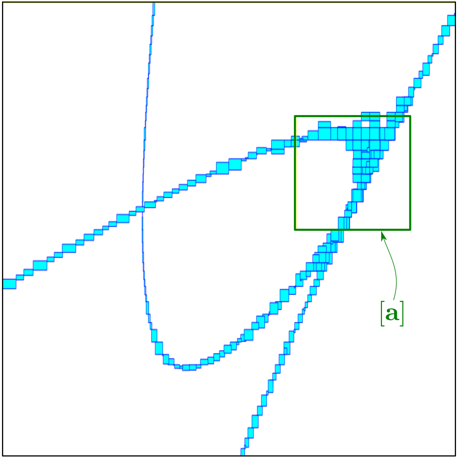
With $[p_1] = [0, 2.5]$, $[p_2] = [1, 4]$, $[\omega] = [0, 10]$, with a Matlab implementation, with a forward-backward contractor, and $\varepsilon = 2^{-8}$, Malti et al. got:



https://youtu.be/DaR2NZZIV10?t=2453



With the basic contractors



With the centered contractor $\varepsilon = 2^{-4}$

tubex-lib
1.0
Search docs
⊟ Tubes: basics
Definition
Arithmetics on tubes
Integrals of tubes
Set-inversion
Contractors for tubes
Implementation
Installing the Tubex library
How to handle tubes with Tubex
Graphical tools
Examples

The following is an extract from the paper Guaranteed computation of robot trajectories.

Definition

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

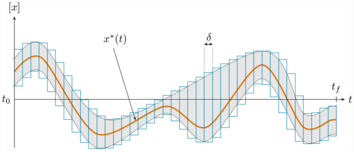
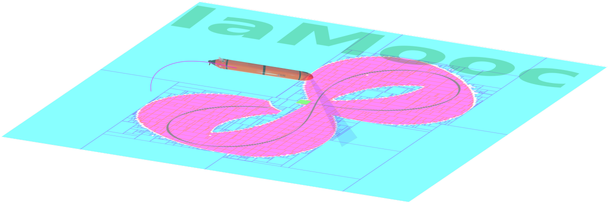


Fig. 1 A tube $[x](\cdot)$ represented by a set of slices. This representation can be used to enclose signals such as $x^*(\cdot)$.

Code example:

```
float timestep = 0.1;
Interval domain(0,10);
Tube x(domain, timestep, Function("t", "(t-5)^2 + [-0.5,0.5]"));
```

http://www.codac.io/



https://www.ensta-bretagne.fr/iamooc/

References

1. Interval analysis [10, 7]
2. Codac [12]
3. Separators [6]
4. Bode plot [2]
5. Example of Ackermann [1]
6. Robust control problem [5][11]
7. Turkulov system [9][4]
8. Code associated with the examples [3]
9. IAMOOC [8]

J. Ackermann, H. Hu, and D. Kaesbauer.
Robustness analysis: a case study.
"IEEE Transactions on Automatic Control", 35(3):352–356, 1990.

M. Dao, M. Di-Loreto, L. Jaulin, J. Lafay, and J. Loiseau.
Application des méthodes intervalles aux systÃšmes à retards.
In *CIFA2004 (Conférence Internationale Francophone d'Automatique), In CDROM*, Douz (Tunisie), 2004.

L. Jaulin.
Codes associated with the talk 'interval analysis with application to robust control of linear systems' at the winter school in lille.

L. Jaulin.
Asymptotically minimal contractors based on the centered form; application to the stability analysis of linear systems.

arXiv:2307.10502, math.NA, 2023.

- L. Jaulin, I. Braems, and E. Walter.
  Interval methods for nonlinear identification and robust control.
  In *In Proceedings of the 41st IEEE Conference on Decision and Control (CDC)*, 9-13 decembre 2002, Las Vegas, 2002.

- L. Jaulin and B. Desrochers.
  Introduction to the algebra of separators with application to path planning.
  Engineering Applications of Artificial Intelligence, 33:141–147, 2014.

- L. Jaulin, M. Kieffer, O. Didrit, and E. Walter.
  Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics.
  Springer-Verlag, London, 2001.

L. Jaulin, O. Reynet, B. Desrochers, S. Rohou, and J. Ninin.
IaMOOC, Interval analysis with applications to parameter estimation and robot localization ,
www.ensta-bretagne.fr/iamooc/.
ENSTA-Bretagne, 2019.

R. Malti, M. Rapaić, and V. Turkulov.
A unified framework for robust stability analysis of linear irrational systems in the parametric space.
Automatica, 2022.
Second version, under review (see also https://hal.archives-ouvertes.fr/hal-03646956).

R. Moore.
Methods and Applications of Interval Analysis.
Society for Industrial and Applied Mathematics, jan 1979.

A. Rauh.

Sensitivity Methods for Analysis and Design of Dynamic Systems with Applications in Control Engineering.
Shaker–Verlag, 2017.

S. Rohou.
Codac (Catalog Of Domains And Contractors), available at `http://codac.io/`.
Robex, Lab-STICC, ENSTA-Bretagne, 2021.