Contractors and Linear Matrix Inequalities

Jeremy Nicola and Luc Jaulin

Abstract. Linear Matrix Inequalities (LMI for short) is a large class of convex constraints. Boxes, Ellipsoids, linear constraints, can be represented by LMIs. Intersection of LMIs are also LMIs. Interior point methods are able to minimize or maximize any linear criterion of LMIs with a complexity which is polynomial regarding to the number of variables. As a consequence, as shown in this paper, it is possible to build optimal contractors for sets represented by LMIs. When solving a set of nonlinear constraints, one may extract from all constraints those that are LMIs in order to build a single optimal LMI contractor. A combinaison with all contractors obtained for other non LMI constraints can thus be performed upto the fixed point. The resulting propogation is shown to be more efficient than other conventional contractor-based approaches.

Keywords. Interval Analysis, Linear Matrix Inequality, Contractor Programming, Convex Optimization, Constraint Propagation, Robotics.

1. Introduction

Contractor techniques is the approach of combining Interval Analysis [1] [2] [3] and fixed-point [4] methods to solve difficult nonlinear problems such as global optimization or parameter estimation in a guaranteed manner. The collaboration of the two approaches makes it possible to treat high dimensional, nonlinear problems efficiently, and has been used to solve various problems ranging from controller design [4], robot localization and environment mapping [5] to global optimization [6].

While Interval Analysis enables us to deal with nonlinearities and roundoff errors in a guaranteed way, its core algorithms, by bisecting the search space and testing each bisection individually, has a complexity that is exponential with respect to the dimension of the problems treated [7]. This makes this approach well suited for low dimensional problems, but doesn't scale well for higher dimensional problems. On the other hand the goal of Constraints Propagation methods is to reduce the search-space as much as possible without losing any feasible solution using fast algorithms. When combined, those features enable us to deal efficiently with high-dimensional, nonlinear and nonconvex problems in a guaranteed manner, using bisections only as a last resort. In this perspective, we understand why having polynomial time algorithms to contract optimally (i.e. to find the smallest box satisfying the given set of constraints) the domains of interest is so important.

Using well-known convex optimization methods, which have already been used to contract optimally an interval matrix for the semi-definite positive constraint [8], this article will introduce a new optimal contractor for the LMI constraint, somehow generalizing previous work [9] which proposed to apply a local solver dedicated to linear systems, for which, to the best of our knowledge, no optimal algorithms have been presented yet.

The paper is organized as follows. We shall first recall what a Linear Matrix Inequality (LMI) is and present a few problems that can be represented as LMIs. In the third section we will present the problem of convex optimization under LMI constraints, and see how it can be used in the context of Interval Analysis. In the fourth section we will make a brief recall on Interval Constraints Propagation and contractor programming and show how to leverage optimization under LMI constraints methods to create a minimal contractor for the LMI constraint. Section 5 will recall the branch and bound like algorithm used to pave sets using contractors. In section 6 we will present some examples of application, compare the efficiency of our contractor with existing ones, and illustrate the power of contractor programming by mixing convex and nonconvex constraints. Section 7 will conclude the paper.

2. Linear Matrix Inequalities

In this section, we will recall some properties about LMIs. For an in depth review on the topic, see [10] and [11].
A Linear Matrix Inequality (LMI) has the form:

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + x_1\mathbf{F}_1 + x_2\mathbf{F}_2 + \cdots + x_n\mathbf{F}_n \succeq 0$$

with $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ the unknown, $\mathbf{F}_{i=0,\ldots,n} \in \mathbb{R}^{n\times n}$ a set of symmetric matrices.
The inequality means that $\mathbf{F}(\mathbf{x})$ is a positive definite matrix, i.e.:

$$\forall\, \mathbf{u} \;\in \mathbb{R}^n \mathbf{u}^T\mathbf{F}\mathbf{u} > \mathbf{0}\;, \mathbf{u} \neq \mathbf{0}.$$

Since the set

$$\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{F}(\mathbf{x}) \succeq 0\}$$

is convex, an LMI is a convex constraint on **x**.
A set of linear inequalities is an LMI. For example

$$\begin{cases} I_1(\mathbf{x}) = & a_{11}x_1 + a_{12}x_2 + b_1 & \geq 0 \\ I_2(\mathbf{x}) = & a_{21}x_1 + a_{22}x_2 + b_2 & \geq 0 \end{cases}$$

is equivalent to the following LMI:

$$\begin{pmatrix} I_1(\mathbf{x}) & 0 \\ 0 & I_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + b_1 & 0 \\ 0 & a_{21}x_1 + a_{22}x_2 + b_2 \end{pmatrix} \succeq \mathbf{0}$$

i.e.,

$$\begin{pmatrix} b_1 & 0 \\ 0 & b_2 \end{pmatrix} + x_1 \begin{pmatrix} a_{11} & 0 \\ 0 & a_{21} \end{pmatrix} + x_2 \begin{pmatrix} a_{12} & 0 \\ 0 & a_{22} \end{pmatrix} \succeq \mathbf{0}$$

A box is described by an LMI. $\mathbf{x} \in [\mathbf{x}]$, with $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$,
$[\mathbf{x}] = [\underline{x}_1, \bar{x}_1] \times [\underline{x}_2, \bar{x}_2] \times \cdots \times [\underline{x}_n, \bar{x}_n] \in \mathbb{IR}^n$, where $\mathbb{IR}^n$ is the set of all the intervals of $\mathbb{R}^n$, is an LMI. Indeed, it can be decomposed as a set of $2 \times n$ linear inequalities:

$$\begin{cases} \mathbf{I}_1^-(\mathbf{x}) = & -\underline{x}_1 + x_1 & \geq 0 \\ \mathbf{I}_1^+(\mathbf{x}) = & \bar{x}_1 - x_1 & \geq 0 \\ \mathbf{I}_2^-(\mathbf{x}) = & -\underline{x}_2 + x_2 & \geq 0 \\ \mathbf{I}_2^+(\mathbf{x}) = & \bar{x}_2 - x_2 & \geq 0 \\ & \vdots & \\ \mathbf{I}_n^-(\mathbf{x}) = & -\underline{x}_n + x_n & \geq 0 \\ \mathbf{I}_n^+(\mathbf{x}) = & \bar{x}_n - x_n & \geq 0 \end{cases}$$

which in turn can be formulated as an LMI. For example, the box

$$[\mathbf{x}] = [-10, 15] \times [3, 7]$$

is an LMI and is written as:

$$\mathbf{B}(x) = \begin{pmatrix} x_1 + 10 & 0 & 0 & 0 \\ 0 & 15 - x_1 & 0 & 0 \\ 0 & 0 & x_2 - 3 & 0 \\ 0 & 0 & 0 & 7 - x_2 \end{pmatrix} \succeq \mathbf{0}$$

i.e.,

$$\mathbf{B}(\mathbf{x}) = \begin{pmatrix} -\underline{x}_1 & 0 & 0 & 0 \\ 0 & \bar{x}_1 & 0 & 0 \\ 0 & 0 & -\underline{x}_2 & \\ 0 & 0 & 0 & \bar{x}_2 \end{pmatrix} + x_1 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \succeq \mathbf{0}$$

An ellipsoid in $\mathbb{R}^d$ is described by an LMI. An ellipsoid $\mathbb{E}$ in $\mathbb{R}^d$ is described as:

$$\mathbb{E} = \{\mathbf{x} \in \mathbb{R}^d \mid r - (\mathbf{x} - \mathbf{c})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{c}) \geq 0\}$$

whith $r > 0$ its radius, $\mathbf{c} \in \mathbb{R}^d$ its center and $\mathbf{P}$ its characteristic matrix.

Using Schur complement theorem, which states that for a set of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ respectively of dimensions $p \times p, p \times q, q \times p, q \times q$,

$$\left\{ \begin{array}{cc} \mathbf{C} & \succ \mathbf{0} \\ \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C} & \succeq \mathbf{0} \end{array} \right. \Leftrightarrow \left(\begin{array}{cc} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{array} \right) \succeq \mathbf{0}$$

we show that $\mathbb{E}$ can be described as the LMI:

$$\mathbf{E}(\mathbf{x}) = \begin{pmatrix} r & \mathbf{x} - \mathbf{c}^T \\ \mathbf{x} - \mathbf{c} & \mathbf{P} \end{pmatrix} \succeq \mathbf{0}$$

For example, the unit disk $\mathbb{D}$ in $\mathbb{R}^2$ is an ellipse with a characteristic matrix $\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, a radius $r = 1$, centered on the origin $\mathbf{c} = (0, 0)^T$:

$$\mathbb{D} = \{\mathbf{x} \in \mathbb{R}^2 \mid 1 - \mathbf{x}^T \mathbf{Id}^{-1} \mathbf{x} \geq 0\}$$

i.e.,

$$1 - x_1^2 - x_2^2 \geq 0$$

Using the Schur complement theorem with

$$\mathbf{A} = 1, \mathbf{B} = \mathbf{C}^T = \begin{pmatrix} x_1 & x_2 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

we get the LMI:

$$\begin{pmatrix} 1 & x_1 & x_2 \\ x_1 & 1 & 0 \\ x_2 & 0 & 1 \end{pmatrix} \succeq \mathbf{0}$$

i.e.,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + x_1 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \succeq \mathbf{0}$$

Many other convex sets can be represented by LMIs. See [12, 11] for a presentation of a consequent amount of LMI-representable sets.

A distance constraint is partly an LMI constraint. Given two points $\mathbf{x}_1, \mathbf{x}_2$ in $\mathbb{R}^d$, and y in $\mathbb{R}$ the distance between $\mathbf{x}_1, \mathbf{x}_2$ such that

$$y^2 = (x_1^1 - x_2^1)^2 + (x_1^2 - x_2^2)^2 + \cdots + (x_1^d - x_2^d)^2$$

This constraint is partly an LMI constraint. Indeed, we have:

$$y^2 = (\mathbf{x}_1 - \mathbf{x}_2)\mathbf{Id}(\mathbf{x}_1 - \mathbf{x}_2) \Leftrightarrow \begin{cases} y^2 \geq (\mathbf{x}_1 - \mathbf{x}_2)\mathbf{Id}(\mathbf{x}_1 - \mathbf{x}_2) \; (1) \\ y^2 \leq (\mathbf{x}_1 - \mathbf{x}_2)\mathbf{Id}(\mathbf{x}_1 - \mathbf{x}_2) \; (2) \end{cases}$$

Thanks to the Schur complement theorem recalled above, we can reframe (1) as

$$\mathbf{R}(\mathbf{x}) = \begin{pmatrix} y^2 & \mathbf{x}_1 - \mathbf{x}_2{}^T \\ \mathbf{x}_1 - \mathbf{x}_2 & \mathbf{Id} \end{pmatrix} \succeq \mathbf{0}$$

Which is an LMI constraint. On the other hand, (2) is not a convex constraint and can't be framed as an LMI.

A system described by several LMIs can be described by a single LMI. The set $\mathbb{S}_{12} = \mathbb{S}_1 \cap \mathbb{S}_2$ with $\mathbb{S}_1, \mathbb{S}_2$ two convex sets described by the LMIs $\mathbf{S}_1(\mathbf{x}) \succeq \mathbf{0}, \mathbf{S}_2(\mathbf{x}) \succeq \mathbf{0}$ is convex and is also described as an LMI $\mathbf{S}_{12}(\mathbf{x}) \succeq \mathbf{0}$, obtained by concatenation of $\mathbf{S}_1(\mathbf{x})$ and $\mathbf{S}_2(\mathbf{x})$:

$$\mathbf{S}_{12}(\mathbf{x}) = \mathrm{diag}(\mathbf{S}_1(\mathbf{x}), \mathbf{S}_2(\mathbf{x})) = \begin{pmatrix} \mathbf{S}_1(\mathbf{x}) & 0 \\ 0 & \mathbf{S}_2(\mathbf{x}) \end{pmatrix} \succeq \mathbf{0}$$

3. Optimization under LMI constraints

Given a vector of variables $\mathbf{x} \in \mathbb{R}^n$, a cost vector $\mathbf{c} \in \mathbb{R}^n$ and a matrix $\mathbf{F}$, an optimization problem under the LMI constraint $\mathbf{F}(\mathbf{x}) \succeq \mathbf{0}$ is stated as follows:

$$P_{\mathbf{c},\mathbf{F}(\mathbf{x})} = \begin{cases} \text{minimize:} & \mathbf{c}^T\mathbf{x} \\ \text{under the constraint:} & \mathbf{F}(\mathbf{x}) \succeq \mathbf{0} \end{cases}$$

Many engineering problems, for example in the field of control theory [13], can be framed as an optimization problem under LMI constraints. The book [11] presents a consequent amount of problems that can be formulated as optimization problems under LMI constraints. Only trivial cases of optimization under LMI constraints can be solved analytically. On the other hand, numerical methods have been developped to solve this type of problem efficiently.

Among such methods, the most efficient are based on interior point methods. Interior point methods were introduced in 1984 by Karmarkar for solving linear programming problems with a polynomial-time algorithm [14]. A lot of research activities followed, leading to [15], where general interior-point methods solving LMI problems are introduced. We refer the reader to [10] for a more detailed review on the algorithms for solving LMI optimization problems and their implementation.

Unlike simplex method commonly used in linear programming, which moves along the edges of the polytope defining the feasible set until it reaches the vertex of optimum solution, interior point methods start from a feasible solution and interates inside this set until the optimal solution is found.

Computing the box-hull of a set in $\mathbb{R}^d$ described by an LMI reduces to solving $2d$ optimization problems under LMI constraints. The box-hull, or the bounding-box $[\mathbb{X}]$ of a set $\mathbb{X}$ in $\mathbb{R}^d$ is the smallest axis-aligned box enclosing $\mathbb{X}$.

Indeed, for each dimension d we are looking for the maximal and minimal values $\bar{x}_d, \underline{x}_d$ of x_d, giving us the interval $[x_d] = [\underline{x}_d, \bar{x}_d]$, their Cartesian product $[\mathbf{x}] = [x_1] \times [x_2] \cdots \times [x_d]$ being the box-hull of $\mathbb{X}$.

Since finding these $2 \times d$ extrema reduces to solving an optimization problem under LMI constraints for each of them, each minimization problem having a polynomial-time complexity, the box-hull problem also has a polynomial-time complexity.

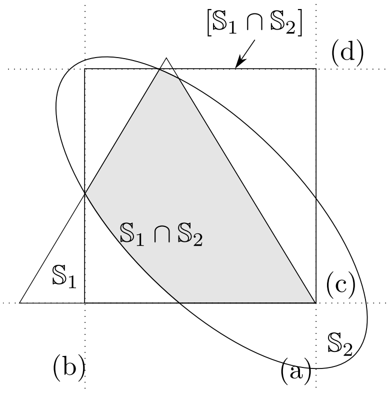

$[\mathbb{S}_1 \cap \mathbb{S}_2]$
(d)
$\mathbb{S}_1 \cap \mathbb{S}_2$
$\mathbb{S}_1$
(c)
$\mathbb{S}_2$
(b)
(a)

FIGURE 1. Characterization of $[\mathbb{S}_1 \cap \mathbb{S}_2]$ (a) maximization of x_1, (b) minimization of x_1, (c) minimization of x_2, (d) maximization of x_2

TABLE 1. Properties of contractors

Possible properties of a contractor $\mathbf{C}_{\mathbb{X}}$	
Monotonicity	$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathbf{C}_{\mathbb{X}}([\mathbf{x}]) \subset \mathbf{C}_{\mathbb{X}}([\mathbf{y}])$
Minimality	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{C}_{\mathbb{X}}([\mathbf{x}]) = [[\mathbf{x}] \cap \mathbb{X}]$
Thinness	$\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{C}_{\mathbb{X}}(\{\mathbf{x}\}) = \{\mathbf{x}\} \cap \mathbb{X}$
Idempotence	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{C}_{\mathbb{X}}(\mathbf{C}_{\mathbb{X}}([\mathbf{x}])) = \mathbf{C}_{\mathbb{X}}([\mathbf{x}])$
Convergence	$[\mathbf{x}](k) \rightarrow \mathbf{x} \Rightarrow \mathbf{C}_{\mathbb{X}}([\mathbf{x}](k)) \rightarrow \{\mathbf{x}\} \cap \mathbb{X}$

Illustration. Given two sets of $\mathbb{R}^2$: a triangle $\mathbb{S}_1$ and an ellipse $\mathbb{S}_2$ described by the LMIs $\mathbf{S}_1(\mathbf{x}) \succeq \mathbf{0}, \mathbf{S}_2(\mathbf{x}) \succeq \mathbf{0}$,, we want to find the box-hull $[\mathbb{S}_\cap]$ for their intersection $\mathbb{S}_\cap = \mathbb{S}_1 \cap \mathbb{S}_2$, described by the LMI $\mathbf{S}(\mathbf{x}) = \begin{pmatrix} \mathbf{S}_1(\mathbf{x}) & 0 \\ 0 & \mathbf{S}_2(\mathbf{x}) \end{pmatrix}, \succeq \mathbf{0}$.
$\mathbb{S}_\cap, \mathbb{S}_1$ and $\mathbb{S}_2$ are illustrated on figure 1.

First, we are looking for an upper-bound $\bar{x}_1$ for x_1. Therefore we solve the optimization problem:

$$P_{\mathbf{c},\mathbf{S}(\mathbf{x})} = \begin{cases} \text{minimize:} & \mathbf{c}^T\mathbf{x} \\ \text{under the constraint:} & \mathbf{S}(\mathbf{x}) \succeq \mathbf{0} \end{cases}$$

with $\mathbf{c}^T = \begin{pmatrix} -1 & 0 \end{pmatrix}$. The dashed-line (a) illustrates the solution found by the algorithm while searching for $\bar{x}_1$. We are then looking for the lower-bound $\underline{x}_1$ for x_1, which is done by solving $P_{\mathbf{c},\mathbf{S}(\mathbf{x})}$ with $\mathbf{c}^T = \begin{pmatrix} 1 & 0 \end{pmatrix}$. The solution is the dashed-line (b). The same process is then repeated for x_2 as illustrated on the dashed lines (c), (d).
The box-hull $[\mathbb{S}]$ is then given by the Cartesian product $[x_1] \times [x_2]$ of the intervals $[x_1] = [\underline{x}_1, \bar{x}_1], [x_2] = [\underline{x}_2, \bar{x}_2]$

4. Contractors

An operator $\mathbf{C}_{\mathbb{X}} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ is a contractor for the set $\mathbb{X}$ if:

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

A contractor associated to a constraint is a contractor associated with the set $\mathbb{X}$ of all x which satisfy the constraint.

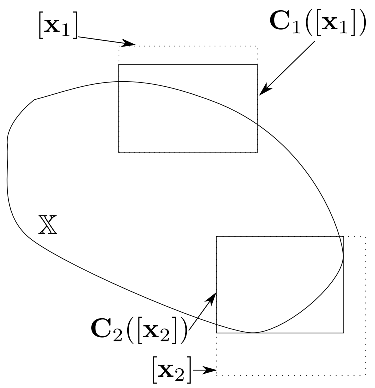

$[\mathbf{x}_1]$
$\mathbf{C}_1([\mathbf{x}_1])$
$\mathbb{X}$
$\mathbf{C}_2([\mathbf{x}_2])$
$[\mathbf{x}_2]$

FIGURE 2. Illustration of a minimal and non-minimal contractor for the set $\mathbb{X}$: $\mathbf{C}_2$ is minimal, $\mathbf{C}_1$ is not

Some properties of contractors are recalled in table 1. The minimality of a contractor is a desirable property. Indeed, if a contractor is minimal, it will remove all the infeasible solutions and return the box-hull of the feasible solutions. Figure 2 illustrates the minimality of a contractor.

Besides, since an optimal contractor is idempotent, we only need to apply it once to reach the fixpoint, which can save us a considerable amount of processing time.

A classical way to contract a box with respect to a constraint is to use the forward-backward algorithm [16].
For example, consider the following constraint:

$$(x_1 + x_2).x_3 \in [1, 2]$$

The forward-backward contractor associated with this constraint is depicted in the algorithm 1.

Algorithm 1 $\mathbf{C}([x_1] \times [x_2] \times [x_3])$ for the constraint $(x_1 + x_2).x_3 \in [1, 2]$

```
1: $[a] = [x_1] + [x_2]$      // Decomposition into
2: $[b] = [a].[x_3]$          // elementary constraints
3: $[b] = [b] \cap [1, 2]$    // Forward propagation
4: $[x_3] = [x_3] \cap \frac{[b]}{[a]}$    // Backward propagations
5: $[a] = [a] \cap \frac{[b]}{[x_3]}$
6: $[x_1] = [x_1] \cap [a] - [x_2]$
7: $[x_2] = [x_2] \cap [a] - [x_1]$
```

Lines 1 and 2 decompose the constraint into elementary constraints. Line 3 is the forward propagation step, and lines 4 to 7 are the backward propagation steps.
Now if we apply this contractor on the box $[\mathbf{x}] = [-\infty, 10] \times [1, 3] \times [8, 9]$ for example, it gets contracted to $[\mathbf{x}'] = [-2.88889, -0.75] \times [1, 3] \times [8, 9]$.
Optimal contractor under LMI constraints. Let $\mathbb{S}$ be a convex set described by the LMI $\mathbf{S}(\mathbf{x}) \succeq \mathbf{0}$, and $[\mathbf{x}] \in \mathbb{IR}^n$ a box described by the LMI $\mathbf{B}(\mathbf{x}) \succeq \mathbf{0}$. The operator $\mathbf{C}_{\text{LMI}}$, which maps $[\mathbf{x}]$ to the box-hull of the set $\mathbb{S}^{\cap}$ described by $\begin{pmatrix} \mathbf{B}(\mathbf{x}) & 0 \\ 0 & \mathbf{S}(\mathbf{x}) \end{pmatrix} \succeq \mathbf{0}$ is a contractor for $\mathbb{S}$, and it is minimal.

Implementing the LMI contractor. Using the C++ library lib-SDPA [17], we implemented the LMI contractor for the IBEX library [18] [19] [20]. Given a set of matrices $\mathbf{F}_0, \mathbf{F}_1, \dots \mathbf{F}_n$ and a box $[\mathbf{x}] \in \mathbb{IR}^n$ as inputs, it contracts $[\mathbf{x}]$ with respect to the LMI constraint:

$$\mathbf{C}_{\text{LMI}}(\mathbf{x}) = \text{diag}(\mathbf{B}(\mathbf{x}), \mathbf{F}_0 + \sum_{i=1}^{n} \mathbf{F}_i(\mathbf{x}_i)) \succeq \mathbf{0}$$

where $\mathbf{B}(\mathbf{x}) \succeq \mathbf{0}$ is the LMI constraint for $\mathbf{x} \in [\mathbf{x}]$ as presented in section 2. It should be noted that each time an LMI contractor is called, since the bounds of the box to be contracted are possibly changed, we have to partly rebuild the contractor. This requires to modify the $\mathbf{F}_0$ matrix for the box LMI $\mathbf{B}(\mathbf{x})$, which makes it a lightweight operation.

It should also be noted that the current implementation of the LMI contractor has no numerical guarantee. We plan to implement a verification step using [21] and integrate the contractor in the IBEX library.

5. Paving with contractors

We want a paving $(\mathbb{I}, \mathbb{B}, \mathbb{O})$ of a set $\mathbb{S}$ described by a set of constraints, that is, a set of non-overlapping boxes that can belong to three categories:

- $\mathbb{I}$, which means the box belongs to $\mathbb{S}$
- $\mathbb{O}$, which means the box does not belong to $\mathbb{S}$
- $\mathbb{B}$, which means we can't say for sure if the box belongs to $\mathbb{I}$ or $\mathbb{O}$

Building the lists defining the sets $\mathbb{I}, \mathbb{B}, \mathbb{O}$ involves three core steps. First, the box $[\mathbf{x}]$ representing the initial domain is contracted to a box $[\mathbf{x}']$ by an *inner contractor* $\mathbf{C}_{in}$, that is a contractor that guarantees:

$$[\mathbf{x}'] \subseteq [\mathbf{x}] \wedge [\mathbf{x}] \setminus [\mathbf{x}'] \subseteq \mathbb{I}$$

The boxes resulting from the set-difference $[\mathbf{x}] \setminus [\mathbf{x}']$ are then pushed in the list $\mathbb{I}$.

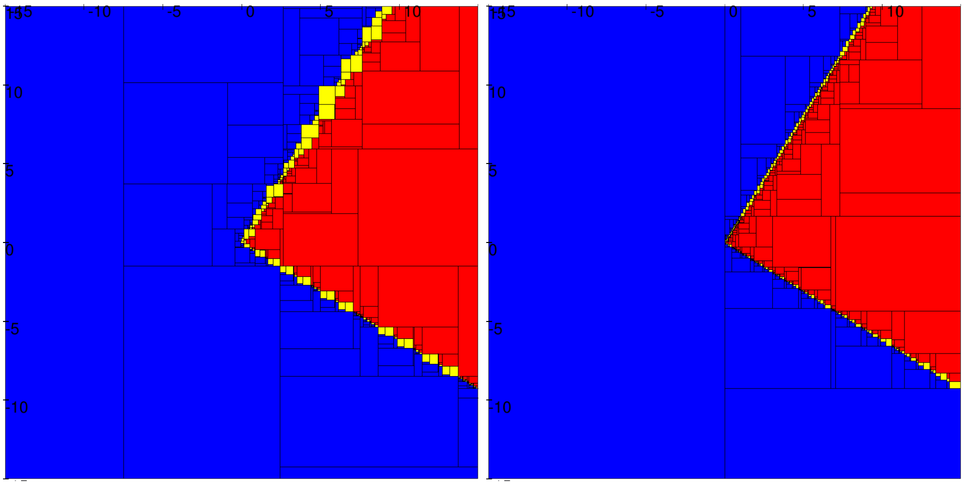
If $[\mathbf{x}'] \neq \emptyset$ then $[\mathbf{x}]'$ is contracted to a box $[\mathbf{x}'']$ by an *outer contractor* $\mathbf{C}_{out}$ that guarantees:

$$[\mathbf{x}]'' \subseteq [\mathbf{x}]' \wedge [\mathbf{x}]' \setminus [\mathbf{x}]'' \subseteq \mathbb{O}$$

Finally, if $[\mathbf{x}]'' \neq \emptyset$ then $[\mathbf{x}]''$ is bisected in two new boxes that are pushed in the list of boundary boxes $\mathbb{B}$. The recusion stops when a specified fixed number of bisections b have been performed. The algorithm 2 summarizes these steps.

Algorithm 2 $(\mathbb{I}, \mathbb{B}, \mathbb{O}) = \textbf{pave}([\mathbf{x}], b)$

```
$b' \leftarrow 0$    // Number of bisections performed
$\mathbb{I} \leftarrow \emptyset, \mathbb{O} \leftarrow \emptyset, \mathbb{B} \leftarrow [\mathbf{x}]$
while $b' \neq b$ do
    $[\mathbf{x}] \leftarrow$ first box in $\mathbb{B}$
    pop $[\mathbf{x}]$ from $\mathbb{B}$
    $[x]' \leftarrow \mathbf{C}_{in}([x]), \mathbb{I} \leftarrow \mathbb{I} \cup ([\mathbf{x}] \setminus [\mathbf{x}]')$ // inner contraction
    $[x]'' \leftarrow \mathbf{C}_{out}([x]'), \mathbb{O} \leftarrow \mathbb{O} \cup ([\mathbf{x}]' \setminus [\mathbf{x}]'')$ // outer contraction
    if $[\mathbf{x}''] \neq \emptyset$ then
        $([\mathbf{x}]_1, [\mathbf{x}]_2) \leftarrow \textbf{bisect}([\mathbf{x}]'')$
        push $[\mathbf{x}]_1$ and $[\mathbf{x}]_2$ in $\mathbb{B}$
        $b' \leftarrow b' + 1$
    end if
end while
```



(a)

(b)

FIGURE 3. Subpaving of $\mathbb{F}$ (a) with forward-backward contractors only, (b) with a forward-backward and a LMI contractor

6. Examples

All the illustrations shown here were made using the VIBEs [22] drawer and its C++ API. We did not include any timing information, and it should be noted that on these problems the forward-backward contractor approach is faster than our proposed approach. However, we anticipate that it could be the other way around when dealing with higher dimensional problems, and this kind of problems should be studied in a further work. In the following examples, we ran algorithm 2 with $b = 100$.

6.1. Representation of a simple LMI-set

Consider the following LMI:

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} x_1 & x_2 \\ x_2 & x_1 + x_2 \end{pmatrix} \succeq \mathbf{0}$$

which means that the two eigenvalues λ_1, λ_2 of $\mathbf{F}$ are real positive, i.e.:

$$\begin{aligned} \lambda_1 &= \tfrac{2x_1+x_2+\sqrt{x_2^2+4x_2^2}}{2} \geq 0 \\ \lambda_2 &= \tfrac{2x_1+x_2-\sqrt{x_2^2+4x_2^2}}{2} \geq 0 \end{aligned}$$

Using set-inversion and contractor programming methods, we want to approximate this LMI-set $\mathbb{F}$ with a subpaving. For this we need two contractors:

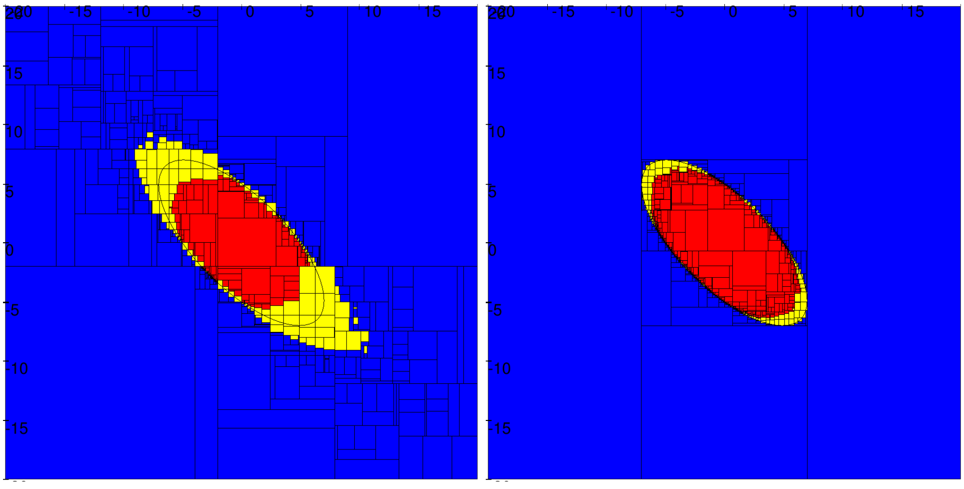
$$\mathbf{C}_{\uparrow\downarrow}^{\text{in}} = \mathbf{C}_{\lambda_1\uparrow\downarrow}^{\text{in}} \cup \mathbf{C}_{\lambda_2\uparrow\downarrow}^{\text{in}}$$

wich will classify in $\mathbb{I}$ parts of the search space that are consistent with the constraints $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ and

$$\mathbf{C}_{\uparrow\downarrow}^{\text{out}} = \mathbf{C}_{\lambda_1\uparrow\downarrow}^{\text{out}} \cap \mathbf{C}_{\lambda_2\uparrow\downarrow}^{\text{out}}$$

wich will classify in $\mathbb{O}$ part of the search space that are inconsistent with the constraints $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$. The boxes classified in $\mathbb{I}$ are drawn in red, the boxes classified in $\mathbb{O}$ are drawn in blue while the undetermined ones are drawn in yellow.

Since the constraint $\mathbf{F}(\mathbf{x}) \succeq \mathbf{0}$ is an LMI constraint, we can replace the outer contractor $\mathbf{C}_{\uparrow\downarrow}^{\text{out}}$ with our LMI contractor $\mathbf{C}_{\text{LMI}}$ introduced in section 4. On the other hand, the constraints defining $\mathbf{C}^{\text{in}}$ are non-convex, and a forward-backward approach is appropriated.



(a)

(b)

FIGURE 4. Characterization of $\mathbb{E}$ (a) with forward-backward contractions only (b) with a forward-backward contractor and the LMI contractor

Figure 3 compares the two approaches, with the same allowed number of bisections. As can be seen, the subpaving computed by the combination of the forward-backward and the LMI contractor is much finer than the one computed with the forward-backward approach only. This is due to the fact that since $\mathbf{C}_{\text{LMI}}$ is minimal, less time is spent bisecting parts of the search space that do not satisfy the positive semidefinite constraint for sure, and more bisections can be performed on elsewhere.

6.2. Characterization of ellipsoidal sets

In this example we want to characterize the set

$$\mathbb{E} = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x}^T \mathbf{P}^{-1} \mathbf{x} \leq r\}$$

$$\Leftrightarrow \mathbb{E} = \{\mathbf{x} \in \mathbb{R}^2 \mid 1.9608x_1^2 + 1.9608x_2^2 - 2.7450x_1x_2 \leq r\}$$

which is an ellipse with $\mathbf{P} = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}$, $r = 5$. As in the previous example, we want to approximate $\mathbb{E}$ with a subpaving. Since it can be described as an LMI-set, we compare the results obtained with the LMI contractor and forward-backward contractor for the outer contractor. The subpavings generated are shown in figure 4. Again, thanks to the minimality of the LMI contractor, we see that the subpaving generated in (b) is much more precise than the one generated in (a) using forward-backward contractors only. However, if $\mathbf{P}$ was diagonal, it should be noted that the forward-backward contractor would be minimal as there would be no multi-occurrences of variables.

6.3. Manipulating the LMI contractor using contractor algebra

In this example we have two sets: a triangle

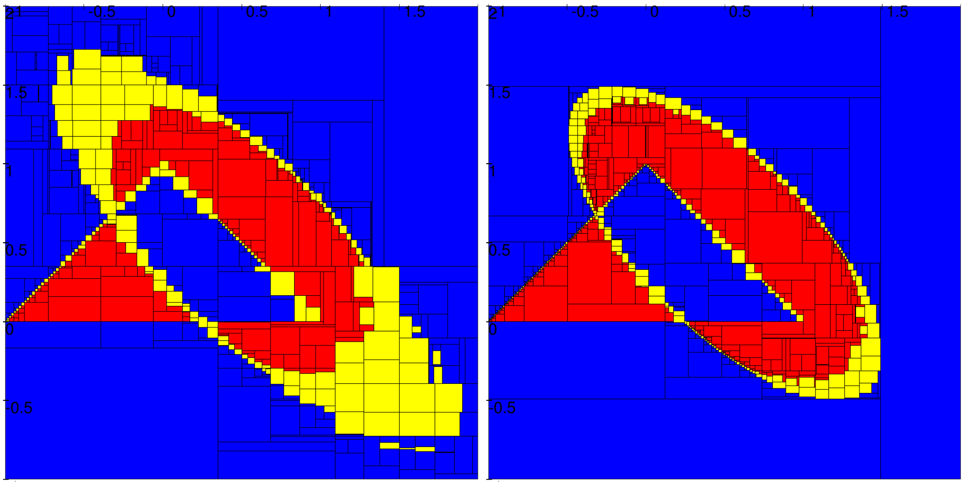
$$\mathbb{T} = \{\mathbf{x} \in \mathbb{R}^2 \mid x_2 + x_1 - 1 \geq 0, x_2 - x_1 - 1 \geq 0, x_2 \geq 0\}$$

and an ellipse:

$$\mathbb{E} = \{\mathbf{x} \in \mathbb{R}^2 \mid (\mathbf{c} - \mathbf{x})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{c}) \leq r\}$$

with $\mathbf{P} = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}$, $r = 0.5$, $\mathbf{c} = \begin{pmatrix} 0.5 & 0.5 \end{pmatrix}^T$.

We would like to compute a subpaving for the set $\mathbb{S} = (\mathbb{T} \cup \mathbb{E}) \setminus (\mathbb{T} \cap \mathbb{E})$.



(a)

(b)

FIGURE 5. Characterization of $\mathbb{S}$ (a) with forward-backward contractions only (b) with forward-backward contractors and LMI contractors

For this we need the inner and outer contractors $\mathbf{C}_{\mathbb{T}}^{\text{in}}, \mathbf{C}_{\mathbb{T}}^{\text{out}}$ for $\mathbb{T}$ and the inner and outer contractors $\mathbf{C}_{\mathbb{E}}^{\text{in}}, \mathbf{C}_{\mathbb{E}}^{\text{out}}$ for $\mathbb{E}$.

Using contractor algebra [19], we obtain the outer and inner contractors for $\mathbb{S}$:

$$\mathbf{C}_{\mathbb{S}}^{\text{out}} = (\mathbf{C}_{\mathbb{E}}^{\text{in}} \cap \mathbf{C}_{\mathbb{T}}^{\text{out}}) \cup (\mathbf{C}_{\mathbb{E}}^{\text{out}} \cap \mathbf{C}_{\mathbb{T}}^{\text{in}})$$

$$\mathbf{C}_{\mathbb{S}}^{\text{in}} = (\mathbf{C}_{\mathbb{E}}^{\text{out}} \cup \mathbf{C}_{\mathbb{T}}^{\text{in}}) \cap (\mathbf{C}_{\mathbb{E}}^{\text{in}} \cup \mathbf{C}_{\mathbb{T}}^{\text{out}})$$

In figure 5 we compare the subpavings obtained when $\mathbf{C}_{\mathbb{E}}^{\text{out}}$ and $\mathbf{C}_{\mathbb{T}}^{\text{out}}$ are forward-backward contractors (a) and when they are LMI contractors. Again, we observe the gain in precision when using the LMI-based contractor.

7. Conclusion and Outlook

In this paper, we introduced a new contractor based on convex optimization under LMI constraints. To the best of our knowledge, this kind of constraint is under-exploited in the interval analysis community and has never been implemented in the contractor framework before. Having LMI constraints in the contractor programming framework has numerous advantages, as it enables us to mix this kind of constraint with other, possibly non-convex constraints, to solve complex problems.

LMI constraints are omnipresent in the context of robotics. For example, range or pseudo-range measurements can be framed as an LMI constraint [23] [24] [25], and therefore the use of our contractor could be used in this kind of application.

Furthermore, in many Bayesian estimation techniques, the parameters to be estimated are modeled as Gaussian random variables, for which the confidence domain is an ellipsoid, which can be represented by an LMI. Therefore we anticipate that our contractor could be used to combine probabilistic and non-probabilistic estimation methods in some possibly interesting ways. The performances of our contractor should also be compared, with timing information, with other state-of-the art constraint propagation based contractors such as HC4, on meaningful, high dimensional problems.

References

[1] Ramon E. Moore. *Interval analysis*. 1966.

[2] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Éric Walter. *Applied interval analysis. With examples in parameter and state estimation, robust control and robotics.* Springer, 2001.

[3] R. B. Kearfott and V. Kreinovich, editors. *Applications of Interval Computations*. Kluwer, Dordrecht, the Netherlands, 1996.

[4] L. Jaulin, I Braems, and E. Walter. Interval methods for nonlinear identification and robust control. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 4, pages 4676–4681 vol.4, Dec 2002.

[5] L. Jaulin. A Nonlinear Set-membership Approach for the Localization and Map Building of an Underwater Robot using Interval Constraint Propagation. *IEEE Transaction on Robotics*, 25(1):88–98, 2009.

[6] Frederic Messine. Deterministic global optimization using interval constraint propagation techniques. *RAIRO - Operations Research - Recherche Opérationnelle*, 38(4):277–293, 2004.

[7] Luc Jaulin and Eric Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, July 1993.

[8] L. Jaulin and D. Henrion. Contracting optimally an interval matrix without loosing any positive semi-definite matrix is a tractable problem. *Reliable Computing*, 11(1):1—17, 2005.

[9] Ignacio Araya, Gilles Trombettoni, and Bertrand Neveu. A contractor based on convex interval taylor. In *Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimzation Problems - 9th International Conference, CPAIOR 2012, Nantes, France, May 28 - June1, 2012. Proceedings*, pages 1–16, 2012.

[10] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[12] A. Ben-Tal and A. Nemirovskii. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, Philadelphia, PA, 2001.

[13] D. Arzelier, B. Clement, and D. Peaucelle. Multi-objective h2/hinfinity/impulse-to-peak control of a space launch vehicle. *European Journal of Control*, 12(1), 2006.

[14] M. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[15] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial methods in convex programming*. SIAM, Philadelphia, PA, 1994.

[16] F. Benhamou, F. Goualard, L. Granvilliers, and J-F. Puget. Revising Hull and Box Consistency. In *ICLP*, pages 230–244, 1999.

[17] Katsuki Fujisawa, Mituhiro Fukuda, Kazuhiro Kobayashi, Masakazu Kojima, Kazuhide Nakata, Maho Nakata, and Makoto Yamashita. Sdpa (semidefinite programming algorithm) user's manual – version 7.0.5. Technical report, 2008.

[18] Ignacio Araya, Gilles Trombettoni, Bertrand Neveu, and Gilles Chabert. Upper bounding in inner regions for global optimization under inequality constraints. *Journal of Global Optimization*, 60(2):145–164, 2014.

[19] Gilles Chabert and Luc Jaulin. Contractor programming. *Artificial Intelligence*, 173(11):1079 – 1100, 2009.

[20] Jordan Ninin, Frédéric Messine, and Pierre Hansen. An Automatic Linear Reformulation Technique Based on Affine Arithmetic. In *International Symposium on Mathematical Programming (ISMP), Chicago, 23/08/2009-28/08/2009*, page 52, http://www.journals.uchicago.edu/, 2009. University of Chicago Press.

[21] Arnold Neumaier and Oleg Shcherbina. Safe bounds in linear and mixed-integer linear programming. *Math. Programming*, pages 283–296, 2004.

[22] Vincent Drevelle and Jeremy Nicola. Vibes: A visualizer for intervals and boxes. *Mathematics in Computer Science*, 8(3-4):563–572, 2014.

[23] L. Jaulin. Range-only SLAM with occupancy maps; A set-membership approach. *IEEE Transaction on Robotics*, 27(5):1004–1010, 2011.

[24] J.L. Blanco, J. Gonzalez, and J.A. Fernández-Madrigal. A pure probabilistic approach to range-only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1436–1441, 2008.

[25] V. Drevelle and P. Bonnifait. High integrity gnss location zone characterization using interval analysis. In *ION GNSS*, 2009.

Jeremy Nicola
Lab STICC - ENSTA Bretagne
2 rue Francois Verny, 29806 Brest Cedex 9, France
e-mail: `jeremy.nicola@ensta-bretagne.org`

Luc Jaulin
Lab STICC - ENSTA Bretagne
2 rue Francois Verny, 29806 Brest Cedex 9, France
e-mail: `luc.jaulin@ensta-bretagne.fr`