Interval centred form for proving stability of non-linear discrete-time systems

Auguste Bourgois

Lab-STICC
ENSTA Bretagne
Brest, France

Forssea Robotics
Paris, France

auguste.bourgois@ensta-bretagne.org

Luc Jaulin

Lab-STICC
ENSTA Bretagne
Brest, France

luc.jaulin@ensta-bretagne.fr

In this paper, we propose a new approach to prove stability of non-linear discrete-time systems. After introducing the new concept of stability contractor, we show that the interval centred form plays a fundamental role in this context and makes it possible to easily prove asymptotic stability of a discrete system. Then, we illustrate the principle of our approach through theoretical examples. Finally, we provide two practical examples using our method : proving stability of a localisation system and that of the trajectory of a robot.

1 Introduction

Proving properties of Cyber Physical Systems (CPS) is an important topic that should be considered when designing reliable systems [2, 7, 20, 28]. Among those properties, *stability* is often wanted for a dynamical system : achieving stability around a given setpoint in the state space of the latter is one of the aims of control theory. Proving stability of a dynamical system can be done rigorously [11], which is of major importance when applied to real-life systems. Indeed, a stable system is considered safe, since its behaviour is predictable.

Let us recall the definition of stability of a dynamical system. Consider the non-linear discrete-time system

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \tag{1}$$

According to Lyapunov's definition of stability [6, 27], the system (1) is *stable* if

$$\forall \varepsilon > 0, \exists \delta > 0, \|\mathbf{x}_0\| < \delta \implies \forall k \geq 0, \|\mathbf{x}_k\| < \varepsilon \tag{2}$$

The system is *asymptotically stable* (or *Lyapunov stable*) if there exists a neighbourhood of $\mathbf{0}$ such that any initial state $\mathbf{x}_0$ taken in this neighbourhood yields a trajectory which converges to $\mathbf{0}$,

$$\exists \delta > 0, \|\mathbf{x}_0\| < \delta \implies \lim_{k \to \infty} \|\mathbf{x}_k\| = 0 \tag{3}$$

Finally, the system is *exponentially stable* if for a given norm $\|\cdot\|$

$$\exists \delta > 0, \exists \alpha \geq 0, \exists \beta \geq 0, \|\mathbf{x}_0\| < \delta \implies \forall k \geq 0, \|\mathbf{x}_k\| \leq \alpha \|\mathbf{x}_0\| e^{-\beta k} \tag{4}$$

A classical method to prove stability of a system is to linearise the latter around $\mathbf{0}$ and check if the eigenvalues are inside the unit disk. However, due the inherent uncertainties of a real-life system's model, no guarantee can be obtained without using interval analysis [25]. The Jury criterion [9] can also

be used on the linearised system in this context, but again, interval computation has to be performed to get a proof of stability [23]. Moreover, to our knowledge, none of the existing methods is able to give an approximation for the neighbourhoods δ and ε used in Lyapunov's definition. Now, finding values for δ and ε is needed in practice, for instance to initialize algorithms which approximate basins of attraction [13, 21, 26] or reachable sets [15].

This paper proposes an original approach to prove Lyapunov stability of a non-linear discrete system, but also to find values for δ and ε. It uses the centred form, a classical concept in interval analysis [16]. Moreover, it does not need the introduction of any Lyapunov function, Jury criterion, linearisation, or any other classical tool used in control theory.

Section 2 briefly presents interval analysis and the notations used in the rest of this paper. Section 3 introduces the new notion of *stability contractor* and gives a theorem which explains why this concept is useful for stability analysis. Section 4 recalls the definition of the centred form. It provides a recursive version of the centred form in the case where the function to enclose is the solution of a recurrence equation. It also shows that the centred form can be used to build stability contractors. Section 5 shows that the approach is able to reach a conclusion for a large class of stable systems and provides a procedure for proving stability. Some examples are given to illustrate graphically the properties of the approach. Section 7 concludes the paper and proposes some perspectives.

2   Introduction to interval analysis

The method presented in this paper is based on interval analysis. In short, interval analysis is a field of mathematics where intervals, *i.e.* connected subsets of $\mathbb{R}$, are used instead of real numbers to perform computations. Doing so allows to enclose all types of uncertainties (from floating-point round-off to modelling errors) of a system and therefore yield a guaranteed enclosure for the solution of a problem related to that system. In this section, we briefly introduce the notations and important concepts used later in this paper. More details about interval analysis and its applications can be found in [10, 18].

An interval is a set delimited by a lower bound x^- and an upper bound x^+ such that $x^- \leq x^+$:

$$[x] = [x^-, x^+]$$

Intervals can be stacked into vectors, and are thus denoted by

$$[\mathbf{x}] = ([x_1], [x_2], \dots, [x_n])$$

We write $[u, v]^{\times n}$ the interval vector of size n, all the components of which are equal to $[u, v]$.

Vectors of intervals are often called *boxes* or *interval vectors*. The set of axis-aligned boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$. Similarly, interval vectors can be concatenated into *interval matrices*.

Intervals, interval vectors (or matrices) can be multiplied by a real λ as such :

$$\lambda [x^-, x^+] = \begin{cases} [\lambda x^-, \lambda x^+] & \text{if } \lambda \geq 0 \\ [\lambda x^+, \lambda x^-] & \text{if } \lambda < 0 \end{cases}$$

We denote by $w([x])$ the width of $[x]$:

$$w([x]) = x^+ - x^-$$

The width of an interval vector $[\mathbf{x}]$ is given by

$$w([\mathbf{x}]) = \max_i (w([x_i]))$$

The absolute value of an interval $[x]$ is

$$|[x]| = \max\left(|x^-|, |x^+|\right)$$

And the norm of an interval vector $[\mathbf{x}]$ is defined as follows

$$\|[\mathbf{x}]\| = \max_i |[x_i]|$$

Later in this paper, we will use the following implication

$$\forall [\mathbf{a}], [\mathbf{b}] \in \mathbb{IR}^n, [\mathbf{a}] \subset [\mathbf{b}] \implies \|[\mathbf{a}]\| < \|[\mathbf{b}]\| \tag{5}$$

The usual arithmetic operators $(+, -, \times, /)$ can be defined over intervals (as well as boxes and interval matrices). Operations involving intervals, interval vectors and interval matrices can therefore be naturally deduced from their real counterpart.

Extending a real function to intervals (and equivalently to interval vectors/matrices) can also be achieved as follows :

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

Except in trivial cases, $f([x])$ usually cannot be written as an interval, whence the use of *inclusion functions*. An inclusion function $[f]([x])$ associated with $f([x])$ yields an interval (or interval vector/matrix) enclosing the set $f([x])$:

$$f([x]) \subset [f]([x])$$

$[f]$ is said to be *minimal* if $[f]([x])$ is the smallest interval (or interval vector/matrix) enclosing the set $f([x])$ (see Figure 1). The minimal inclusion function associated with f will be denoted by $[f([x])]$. An
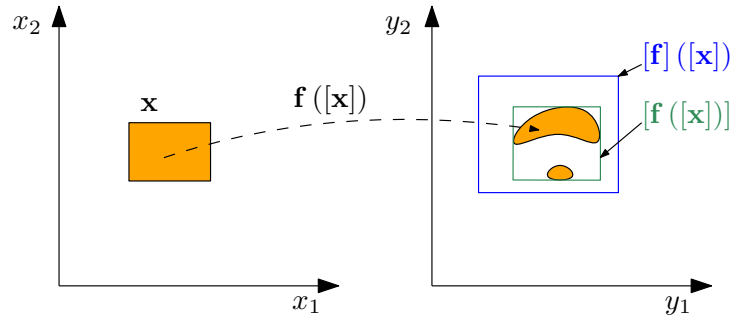

Figure 1: Interval function, inclusion function & minimal inclusion function

inclusion function is said to be *natural* when it is expressed by replacing its variables and its elementary functions and operators by their interval counterparts.

Example 1. Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ such that for $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$

$$f(\mathbf{x}) = \sin(x_1) + \exp(x_2)$$

The natural inclusion function $[f]$ of f is :

$$[f]([\mathbf{x}]) = [\sin]([x_1]) + [\exp]([x_2])$$

where $[\sin]$ and $[\exp]$ are the inclusion functions of sin and exp.

3   Stability contractor

In this section, we present the concept of *stability contractor*, a tool that can be used to rigorously prove the stability of a dynamical system. The rigour of the method comes from the use of interval analysis.

The following new definition adapts the definition of a contractor, as given in [3], to stability analysis.

Definition 1. Consider a box $[\mathbf{x}_0]$ of $\mathbb{R}^n$. A *stability contractor* $\Psi : \mathbb{IR}^n \to \mathbb{IR}^n$ of rate $\alpha < 1$ is an operator which satisfies

$$
\begin{aligned}
&(i) &&[\mathbf{a}] \subset [\mathbf{b}] \implies \Psi([\mathbf{a}]) \subset \Psi([\mathbf{b}]) &&\text{(monotonicity)}\\
&(ii) &&\Psi([\mathbf{a}]) \subset [\mathbf{a}] &&\text{(contractance)}\\
&(iii) &&\Psi(\mathbf{0}) = \mathbf{0} &&\text{(equilibrium)}\\
&(iv) &&\Psi([\mathbf{a}]) \subset \alpha \cdot [\mathbf{a}] \implies \forall k \geq 1, \Psi^k([\mathbf{a}]) \subset \alpha^k \cdot [\mathbf{a}] &&\text{(convergence)}
\end{aligned}
\tag{6}
$$

for all boxes $[\mathbf{a}], [\mathbf{b}]$ inside $[\mathbf{x}_0]$. For $k \geq 1$ Ψ^k denotes the iterated function $\underbrace{\Psi \circ \cdots \circ \Psi}_{k}$. If $k = 0$, Ψ^0 denotes the identity function.

Example 2. If $[x]$ is an interval, the operator $[x] \mapsto [x] \cap 0.9 \cdot [x]$ is a stability contractor whereas the operator $[x] \mapsto 0.9 \cdot [x]$ is not.

Proposition 1. *If Ψ is a stability contractor of rate $\alpha < 1$, then*

$$
\Psi([\mathbf{x}]) \subset \alpha \cdot [\mathbf{x}] \implies \lim_{k \to \infty} \Psi^k([\mathbf{x}]) = \mathbf{0}
\tag{7}
$$

Proof. Let $[\mathbf{x}] \in \mathbb{IR}^n$ and Ψ denote a stability contractor such that $\Psi([\mathbf{x}]) \subset \alpha \cdot [\mathbf{x}]$. Then, according to Equation (6, iv),

$$
\begin{aligned}
\Psi([\mathbf{x}]) \subset \alpha \cdot [\mathbf{x}] &\implies \Psi^k([\mathbf{x}]) \subset \alpha^k \cdot [\mathbf{x}]\\
&\implies \lim_{k \to \infty} \Psi^k([\mathbf{x}]) = \mathbf{0}
\end{aligned}
$$

$\square$

A consequence of this proposition is that getting a stability contractor allows us to prove stability of a system without performing an infinitely long set-membership simulation. It suffices to have one box $[\mathbf{x}] \ni \mathbf{0}$ such that $\Psi([\mathbf{x}])) \subset \alpha \cdot [\mathbf{x}]$, $\alpha < 1$ to conclude that the system is asymptotically stable for all $\mathbf{x} \in [\mathbf{x}]$. In other words, the system is proven to be Lyapunov stable in the neighbourhood $[\mathbf{x}_0]$.

Now, if one can build a stability contractor Ψ for a given dynamical system, then proving Lyapunov stability of the latter for a given initial condition $[\mathbf{x}]$ comes down to applying proposition 1. Building such a contractor is addressed in the next section.

4   Centred form of an interval function

The aim of this section is to present a general method for building a stability contractor for discrete dynamical systems. It is based on the concept on centred form, proposed by Moore in [17]. Additionally, it proposes an algorithm to compute iteratively the centred form of an iterated function.

Consider a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, with a Jacobian matrix $\mathbf{J}(\mathbf{x}) = \frac{d\mathbf{f}}{d\mathbf{x}}(\mathbf{x})$. Consider a box $[\mathbf{x}]$ and one point $\bar{\mathbf{x}}$ in $[\mathbf{x}]$. For simplicity, and without loss of generality, we assume that $\bar{\mathbf{x}} = \mathbf{0} \in [\mathbf{x}]$ and that $\mathbf{f}(\mathbf{0}) = \mathbf{0}$.

Remark 1. If this condition is not satisfied, *i.e.* $\bar{\mathbf{x}} \neq \mathbf{0}$ or $\mathbf{f}(\mathbf{0}) \neq \mathbf{0}$, the problem $\mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in [\mathbf{x}]$ can be transformed into an equivalent problem satisfying the latter :

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \quad \Leftrightarrow \quad \underbrace{\mathbf{y} - \mathbf{f}(\bar{\mathbf{x}})}_{=\mathbf{z}} = -\mathbf{f}(\bar{\mathbf{x}}) + \mathbf{f}(\underbrace{\mathbf{x} - \bar{\mathbf{x}}}_{=\mathbf{p}} + \bar{\mathbf{x}})$$

$$\Leftrightarrow \quad \mathbf{z} = \underbrace{-\mathbf{f}(\bar{\mathbf{x}}) + \mathbf{f}(\mathbf{p} + \bar{\mathbf{x}})}_{=\mathbf{g}(\mathbf{p})} \tag{8}$$

i.e.

$$\begin{cases} \mathbf{y} &=& \mathbf{z} + \mathbf{f}(\bar{\mathbf{x}}) \\ \mathbf{z} &=& \mathbf{g}(\mathbf{p}) \\ \mathbf{p} &=& \mathbf{x} - \bar{\mathbf{x}} \end{cases} \tag{9}$$

Now, consider the new problem $\mathbf{z} = \mathbf{g}(\mathbf{p}), \mathbf{p} \in [\mathbf{p}]$ where $[\mathbf{p}] = [\mathbf{x}] - \bar{\mathbf{x}}$. Since $\mathbf{g}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{0} \in [\mathbf{p}]$, the new problem satisfies the condition stated above.

Let us recall the definition of the centred form, as given by [17] :

Definition 2. The centred form $\mathbf{f}_c$ associated to the function $\mathbf{f}$ is given by

$$[\mathbf{f}_c]([\mathbf{x}]) = ([\mathbf{J}]([\mathbf{x}])) \cdot [\mathbf{x}] \tag{10}$$

where $[\mathbf{J}]$ is the natural extension of $\mathbf{J}$.

Proposition 2. *According to [17], for all* $[\mathbf{x}] \in \mathbb{IR}^n$,

$$\mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}_c]([\mathbf{x}]) \tag{11}$$

Additionally, the centred form tends towards the minimal inclusion function when $[\mathbf{x}]$ *is sufficiently small :*

$$w([\mathbf{f}_c]([\mathbf{x}])) - w([\mathbf{f}([\mathbf{x}])]) = o(w([\mathbf{x}])) \ as \ w([\mathbf{x}]) \to 0 \tag{12}$$

The statements of proposition 2 are illustrated on figure 2.

Theorem 1. *Consider a function* $\mathbf{f}$ *with* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ *and with Jacobian matrix* $\mathbf{J}(\mathbf{x}) = \frac{d\mathbf{f}}{d\mathbf{x}}(\mathbf{x})$. *Denote by* $[\mathbf{f}]$ *and* $[\mathbf{J}]$ *the natural inclusion functions for* $\mathbf{f}$ *and* $\mathbf{J}$. *The centred form* $[\mathbf{f}_c^k]$ *associated to* $\mathbf{f}^k = \mathbf{f} \circ \mathbf{f} \circ \cdots \circ \mathbf{f}$ *is given by the following sequence*

$$\begin{array}{rcl} [\mathbf{z}](0) &=& [\mathbf{x}] \\ [\mathbf{A}](0) &=& Id \\ [\mathbf{z}](k) &=& [\mathbf{f}]([\mathbf{z}](k-1)) \\ [\mathbf{A}](k) &=& [\mathbf{J}]([\mathbf{z}](k-1)) \cdot [\mathbf{A}](k-1) \\ [\mathbf{f}_c^k]([\mathbf{x}]) &=& [\mathbf{A}](k) \cdot [\mathbf{x}] \end{array} \tag{13}$$

Proof. Consider a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ such that $\mathbf{f}(\mathbf{0}) = \mathbf{0}$, and denote $\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}}$ its Jacobian matrix. Write $\mathbf{f}^k$ the k-th iterated function of $\mathbf{f}$. Then for all $\mathbf{x} \in \mathbb{R}^n$, the Jacobian matrix of $\mathbf{f}^k$ is given by :

$$\frac{d\mathbf{f}^k}{d\mathbf{x}}(\mathbf{x}) = \frac{d(\mathbf{f} \circ \mathbf{f}^{k-1})}{d\mathbf{x}}(\mathbf{x}) = \frac{d\mathbf{f}}{d\mathbf{x}}(\mathbf{f}^{k-1}(\mathbf{x})) \cdot \frac{d\mathbf{f}^{k-1}}{d\mathbf{x}}(\mathbf{x}). \tag{14}$$

Define $\mathbf{z}(k) = \mathbf{f}^k(\mathbf{x})$ and $\mathbf{A}(k) = \frac{d\mathbf{f}^k}{d\mathbf{x}}(\mathbf{x})$. Since for all $\mathbf{x}$, $\mathbf{J}(\mathbf{x}) = \frac{d\mathbf{f}}{d\mathbf{x}}(\mathbf{x})$, we get

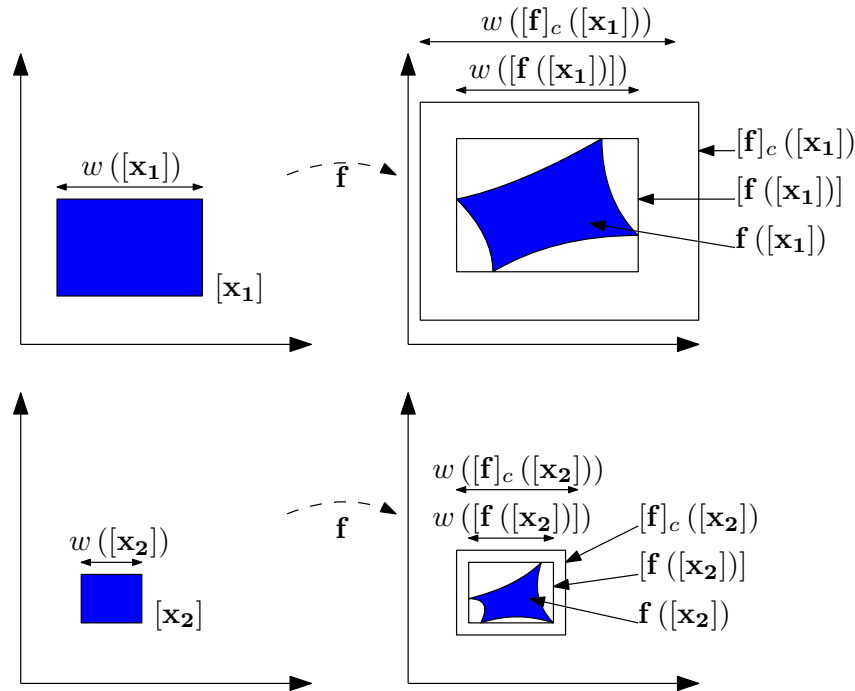$$\mathbf{A}(k) = \mathbf{J}(\mathbf{z}(k-1)) \cdot \mathbf{A}(k-1). \tag{15}$$

Figure 2: Illustration of the centred form extension of an interval function

Now, since $\mathbf{f}(\mathbf{0}) = \mathbf{0}$, it follows that $\mathbf{f}^k(\mathbf{0}) = \mathbf{0}$. Finally, substituting Equation (14) into Equation (10) yields

$$[\mathbf{f}_c^k]([\mathbf{x}]) = \left[\frac{d\mathbf{f}^k}{d\mathbf{x}}\right]([\mathbf{x}]) \cdot [\mathbf{x}]$$

$$= [\mathbf{A}](k) \cdot [\mathbf{x}]$$

$\square$

Remark 2. From Proposition 2, for a given k, we have

$$\lim_{w([\mathbf{x}]) \to 0} \frac{w([\mathbf{f}_c^k]([\mathbf{x}])) - w([\mathbf{f}^k([\mathbf{x}])])}{w([\mathbf{x}])} = 0 \tag{16}$$

which means that $[\mathbf{f}_c^k]$ tends towards the minimal inclusion function when $w([\mathbf{x}]) \to 0$. Now, for a given $[\mathbf{x}]$, even very small, we generally observe the following :

$$\lim_{k \to \infty} \frac{w([\mathbf{f}_c^k]([\mathbf{x}])) - w([\mathbf{f}^k([\mathbf{x}])])}{w([\mathbf{x}])} = \infty. \tag{17}$$

In other words, the pessimism introduced by the centred form increases with k.

Theorem 2. *Consider a function* $\mathbf{f}$ *with* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$*. If there exist a box* $[\mathbf{x}_0] \ni \mathbf{0}$ *and a real* $\alpha < 1$ *such that* $[\mathbf{f}_c]([\mathbf{x}_0]) \subset \alpha \cdot [\mathbf{x}_0]$ *then the interval operator* $\Psi_{\mathbf{f}}([\mathbf{x}]) = [\mathbf{f}_c]([\mathbf{x}])$ *is a stability contractor inside* $[\mathbf{x}_0]$*.*

Proof. Properties (6, i), (6, ii) and (6, iii) of definition 1 are easily checked. Now, let us prove property (6, iv) by induction.

Let $\alpha < 1$ and $[\mathbf{x}] \ni \mathbf{0}$. Property (6, iv) states that

$$[\mathbf{f}_c]([\mathbf{x}]) \subset \alpha[\mathbf{x}] \implies \forall k \geq 0, [\mathbf{f}_c]^k([\mathbf{x}]) \subset \alpha^k \cdot [\mathbf{x}]$$

where $[\mathbf{f}_c]^k = \underbrace{[\mathbf{f}_c] \circ \cdots \circ [\mathbf{f}_c]}_{k}$. Since $[\mathbf{x}] \subset \alpha^0[\mathbf{x}]$, the property holds for $k = 0$.

Let us define the sequence

$$[\mathbf{w}_{k+1}] = [\mathbf{f}_c]([\mathbf{w}_k])$$

Now, assume that the property also holds for a $k > 0$, *i.e.*

$$[\mathbf{w}_k] \subset \alpha^k \cdot [\mathbf{x}]$$

Let us show that

$$[\mathbf{w}_{k+1}] \subset \alpha^{k+1} \cdot [\mathbf{x}]$$

$$
\begin{aligned}
[\mathbf{w}_{k+1}] &= [\mathbf{f}_c]([\mathbf{w}_k]) \\
&= [\mathbf{J}]([\mathbf{w}_k]) \cdot [\mathbf{w}_k] \\
&\subset [\mathbf{J}]([\mathbf{x}]) \cdot [\mathbf{w}_k] && \text{since } [\mathbf{J}] \text{ is inclusion monotonic and } [\mathbf{w}_k] \subset \alpha^k \cdot [\mathbf{x}] \subset [\mathbf{x}] \\
&\subset [\mathbf{J}]([\mathbf{x}]) \cdot (\alpha^k \cdot [\mathbf{x}]) && \text{since } [\mathbf{w}_k] \subset \alpha^k[\mathbf{x}] \\
&= \alpha^k \cdot [\mathbf{J}]([\mathbf{x}]) \cdot [\mathbf{x}] \\
&= \alpha^k \cdot [\mathbf{f}_c]([\mathbf{x}]) \\
&\subset \alpha^k \cdot (\alpha \cdot [\mathbf{x}]) \\
&\subset \alpha^{k+1} \cdot [\mathbf{x}] && \text{since } [\mathbf{f}_c]([\mathbf{x}]) \subset \alpha[\mathbf{x}]
\end{aligned}
$$

$\square$

Theorem 2 shows that the centred form can be used in a very general way to build stability contractors. Other integration methods such as the one proposed by Lohner [14] or the ones based on affine forms [1] do not have this property, even if they usually yield a tighter enclosure of the image set of a function.

5 Proving stability using the centred form

In this section, we show how the centred form can be used to prove stability of non-linear dynamical systems.

5.1 Method

Consider the system described by Equation (1), where $\mathbf{f}(\mathbf{0}) = \mathbf{0}$.

In short, the methods consists in finding an initial box $[\mathbf{x}]$ in the state space of the system such that the centred form of $\mathbf{f}$ is a stability contractor. This implies that iterating this centred form onto that initial box will converge towards $\mathbf{0}$. Since the centred form is an inclusion function for $\mathbf{f}$, that implies that the system will also converge towards $\mathbf{0}$.

More specifically, if for a given box $[\mathbf{x}] \ni \mathbf{0}$ with width δ there exist $q \geq 1$ and $\alpha < 1$ such that $[\mathbf{f}_c^q]([\mathbf{x}]) \subset \alpha \cdot [\mathbf{x}]$, then $[\mathbf{f}_c^q]$ is a stability contractor of rate α (according to theorem 2). Now, since proposition 1 states

$$\lim_{k \to \infty} [\mathbf{f}_c^q]^k([\mathbf{x}]) = \mathbf{0}$$

and since proposition 2 asserts that

$$\forall \mathbf{x} \in [\mathbf{x}], \, \mathbf{f}^q(\mathbf{x}) \in \mathbf{f}^q([\mathbf{x}]) \subset [\mathbf{f}_c^q]([\mathbf{x}])$$

it follows that

$$\forall \mathbf{x} \in [\mathbf{x}], \, \lim_{k \to \infty} (\mathbf{f}^q)^k(\mathbf{x}) = \mathbf{0}$$

which implies asymptotic stability of the system (see Equation 3).

Furthermore, let us define the sequence

$$[\mathbf{x}_{k+1}] = [\mathbf{f}_c^q]^k([\mathbf{x}]) \tag{18}$$

Thus, substituting Equation (18) in property (6, iv) and applying Equation (5) yields

$$\begin{aligned}
\|[\mathbf{x}_{k+1}]\| &\leq \alpha^k \|[\mathbf{x}]\| \\
&\leq \|[\mathbf{x}]\| e^{\ln(\alpha)k} \\
&\leq \|[\mathbf{x}]\| e^{-\beta k}
\end{aligned}$$

where $\beta = -\ln(\alpha) > 0$. This implies exponential stability of the system (see Equation 4).

Our method is summarized by algorithm 1. It takes as inputs the function $\mathbf{f}$ describing the system, the initial box $[\mathbf{x}]$, and a maximum number of iterations N. The latter is necessary, as stated in remark 2 to avoid looping indefinitely. Usually, N does not need to be larger than 10, since the iterated centred form tends to diverge with the number of iterations.

Algorithm 1 Centred form based stability contractor

Input: $\mathbf{f}$, $[\mathbf{x}]$, N
Output: **true** if the system is stable, **false** if undetermined
  $[\mathbf{A}] \leftarrow \mathbf{I}$
  $[\mathbf{z}] \leftarrow [\mathbf{x}]$
  for $i = 1$ **to** N **do**
    $[\mathbf{A}] \leftarrow [\mathbf{J}]([\mathbf{z}]) \cdot [\mathbf{A}]$
    $[\mathbf{z}] \leftarrow [\mathbf{f}]([\mathbf{z}])$
    $[\mathbf{x}_i] \leftarrow [\mathbf{A}] \cdot [\mathbf{x}]$
    if $[\mathbf{x}_i] \subset [\mathbf{x}]$ **then**
      return true
    end if
  end for
  return false

It is important to remember that if the previous algorithm yields a **false** value, that does not necessarily mean that the system is unstable. Indeed, bisecting the initial box $[\mathbf{x}]$ into smaller boxes and running the algorithm on them could prove stability of the system in the neighbourhood defined by those boxes.

5.2   Completeness of the method

Now, let us show that whenever a system actually is exponentially stable, our method will be able to prove it. Given $\eta > 0$, we denote by $\mathscr{B}_\eta$ the set of all hypercubes $[\mathbf{x}]$ centred at $\mathbf{0}$ and such that $w([\mathbf{x}]) \leq \eta$.

Proposition 3. *If the system is exponentially stable around* **0**, *then*

$$\exists \eta > 0, \forall [\mathbf{x}] \in \mathscr{B}_\eta, \exists k > 0, \exists \alpha < 1, [\mathbf{f}_c^k]([\mathbf{x}]) \subset \alpha \cdot [\mathbf{x}]$$

Proof. Let us assume exponential stability of the system described by **f** around **0**, *i.e.* there exists a neighbourhood $\mathscr{N}$ of **0**, such that

$$\exists \beta, 0 < \beta < 1, \exists k > 0, \forall \mathbf{x} \in \mathscr{N}, \|\mathbf{f}^k(\mathbf{x})\|_\infty < \beta \|\mathbf{x}\|_\infty$$

This property translates into

$$[\mathbf{f}^k([\mathbf{x}])] \subset \beta \cdot [\mathbf{x}]$$

for all cubes $[\mathbf{x}]$ in $\mathscr{N}$ centred in **0**, where $[\mathbf{f}^k([\mathbf{x}])]$ is the smallest box which contains the set $\mathbf{f}^k([\mathbf{x}])$. Take one of these cubes $[\mathbf{x}]$ and denote by η its width $w([\mathbf{x}])$. If η is sufficiently small, the pessimism of the centred form becomes arbitrarily small and we have

$$\exists \alpha, \beta \le \alpha < 1, [\mathbf{f}_c^k]([\mathbf{x}]) \subset \alpha[\mathbf{x}]$$

$\square$

6 Application

6.1 Example 1 : Proving stability of the logistic map

The aim of this example is to illustrate the section 5.2. The logistic map is a simple recurrence relation which behaviour can be highly complex or chaotic :

$$x_{k+1} = \rho \cdot x_k \cdot (1 - x_k) \tag{19}$$

The parameter ρ influences the dynamics of the system. For $\rho = 2.4$, the system is stable, but shows an oscillating behaviour around its equilibrium point (see figure 3).

Note that the equilibrium point is not **0** and we thus need to centre our problem to apply the stability method as described in this paper (see remark 1).

Figure 4 displays the behaviour of our algorithm for an initial box $[x_0] = [0.577, 0.585]$. Even if $[x_1]$ is not contained in $[x_0]$, we have $[x_2] \subset [x_0]$. We have thus shown that for an initial state chosen in $[x_0]$, the trajectory will converge towards the stable point.

6.2 Example 2 : Proving stability of a three dimensional map

The aim of this example is to illustrate the steps of algorithm 1 in a higher dimensional problem.

Let us consider the following discrete system:

$$\begin{aligned} \mathbf{x}_{k+1} &= 0.8 \cdot \mathbf{R}\left(\frac{\pi}{6} + x_1, \frac{\pi}{4} + x_2, \frac{\pi}{3} + x_3\right) \cdot \mathbf{x}_k \\ &= \mathbf{f}(\mathbf{x}_k) \end{aligned} \tag{20}$$

where $\mathbf{R}(\varphi, \theta, \psi)$ is the rotation matrix parametrized by roll (φ, around axis X), pitch (θ, around Y) and yaw (ψ, around Z) angles.

With our approach, we show that the discrete system is stable, as depicted by Figure 5 which is a projection of the 3-dimensional system across steps of our algorithm. With $[\mathbf{x}_0] = [-\varepsilon, \varepsilon]^{\times 3}$, where
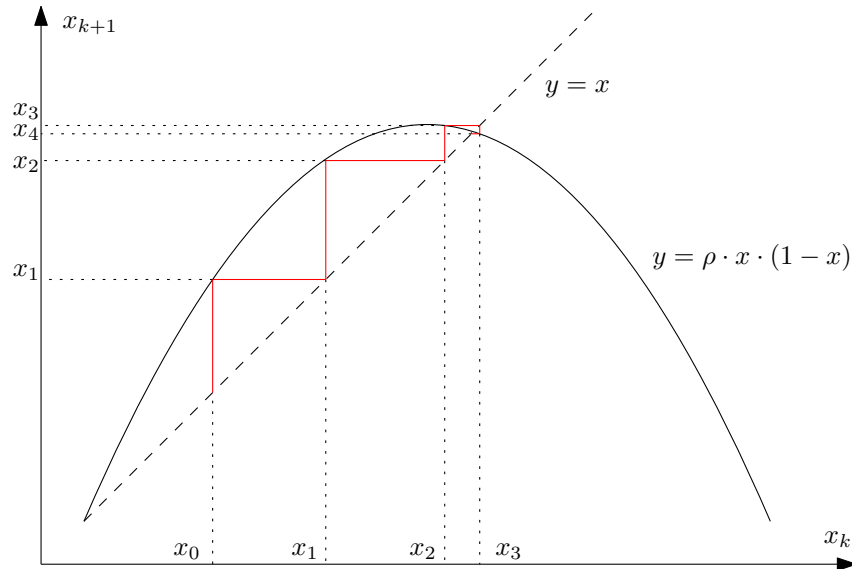
Figure 3: Behaviour of the logistic map for $\rho = 2.4$

$\varepsilon = 0.004$, the algorithm needs 3 iterations to get captured inside the initial box. We thus have proved the stability of the system. The blue sets correspond to the image of $[\mathbf{x}_0]$ by $\mathbf{f}, \mathbf{f}^2, \mathbf{f}^3$ and has been obtained using a Monte-Carlo method for visualization purposes. The point in the centre corresponds to the origin $\mathbf{0}$.

Considering the shape of the set $\mathbf{f}^k([\mathbf{x}_0])$, we understand that zonotope-based methods [5, 29] or Lohner integration methods [14, 30] could get a stronger convergence. However, these efficient operators cannot be used to prove the stability except if we are able to prove that they correspond to a stability contractor, which is not the case yet.

6.3   Example 3 : Validation of a localisation system

The aim of this example is to illustrate how our method could be applied to prove stability of an existing localisation system, before integrating the latter in a robot for example.

Remark 3. Proving the stability of such a localisation system is of importance in robotics. Indeed, the commands are usually computed using the robot's state, estimated by the localisation system. If the latter is not stable, *i.e.* it does not converge towards the actual position of the robot, then control is useless. Note that additionally to stability, accuracy and precision are also wanted features for a localisation system, that we won't address here.

We also briefly explain how our method can be coupled with a paving algorithm to characterize the *stability region* of the localisation system. The latter corresponds to the acceptable set of parameters of the system, in the sense that it remains stable regardless of the parameters picked in that set.

Consider a range-only localisation system using two landmarks $\mathbf{a}$, $\mathbf{b}$ as represented on Figure 6. A static robot is located at position $\mathbf{m}$, and is able to measure the exact distance between itself and the landmarks.

More precisely, we assume that we have the following measurements

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{h}(\mathbf{m}) = \begin{pmatrix} (m_1 - a_1)^2 + (m_2 - a_2)^2 \\ (m_1 - b_1)^2 + (m_2 - b_2)^2 \end{pmatrix}.$$
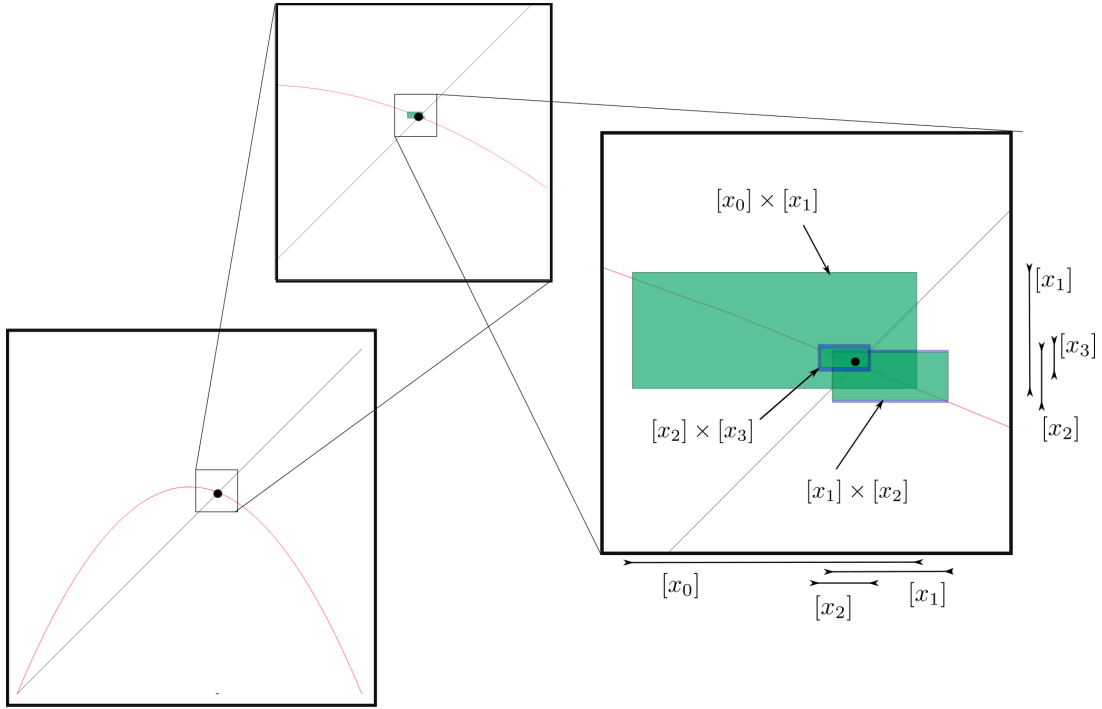
Figure 4: Stability contractor proving stability of the logistic map. The initial box is small, whence the 2 enlargements symbolised by the connected black squares.

Assume also that, at iteration k, the robot believes to be at position $\mathbf{p}_k$ whereas it actually is at position $\mathbf{m}$. We define the Newton sequence as

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{J}_\mathbf{h}^{-1}(\mathbf{p}_k) \cdot (\mathbf{h}(\mathbf{m}) - \mathbf{h}(\mathbf{p}_k))$$

where

$$\mathbf{J}_\mathbf{h}(\mathbf{p}) = 2 \begin{pmatrix} p_1 - a_1 & p_2 - a_2 \\ p_1 - b_1 & p_2 - b_2 \end{pmatrix}$$

We have

$$\mathbf{x}_{k+1} = \underbrace{\mathbf{x}_k + \mathbf{J}_\mathbf{h}^{-1}(\mathbf{x}_k + \mathbf{m}) \cdot (\mathbf{h}(\mathbf{m}) - \mathbf{h}(\mathbf{x}_k + \mathbf{m}))}_{\mathbf{f}(\mathbf{x}_k, \mathbf{m})},$$

where $\mathbf{x}_k = \mathbf{p}_k - \mathbf{m}$ is the localisation error. We want to show that for $\mathbf{m}$ in a given region of the plane, the sequence defined by $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{m})$ is stable, *i.e.*, converges to $\mathbf{0}$.

Remark 4. Of course, this localisation system could be greatly improved. But let us imagine this localisation system has been implemented on a robot and working for years. Its performances are known and trusted, and changing it is not possible or desired. The goal here is to validate the existing system, not to build a new one. The guarantee of stability is a property we can check with our approach.

Now, let us introduce the concept of *stability region*.

Definition 3. The sequence $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{m})$ parametrized by $\mathbf{m} \in [\mathbf{m}]$ is *robustly stable* if

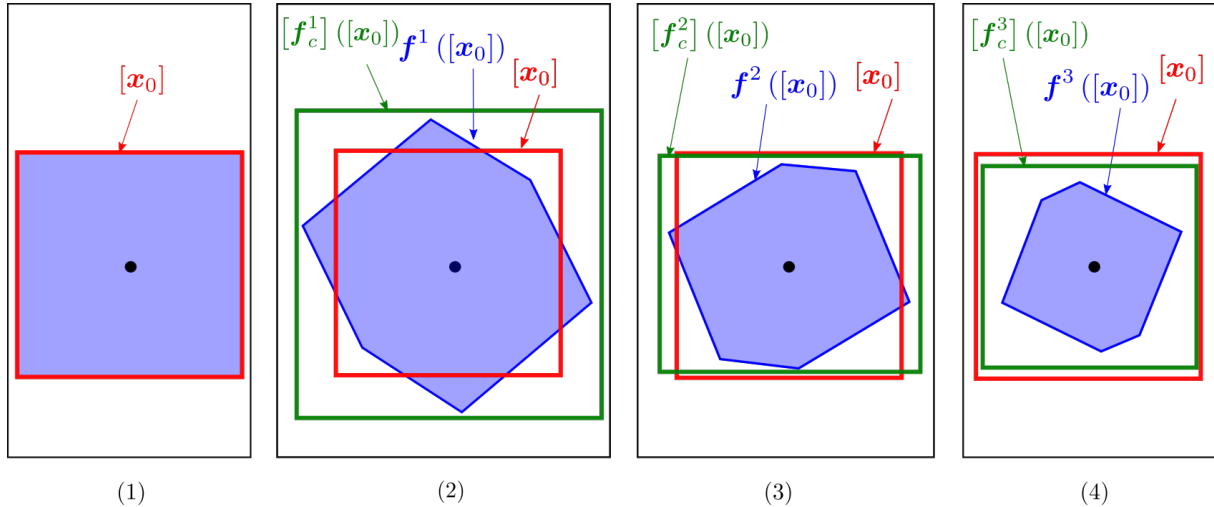$$\forall \mathbf{m} \in [\mathbf{m}], \lim_{k \to \infty} \mathbf{x}_k = \mathbf{0} \tag{21}$$

Figure 5: View in the (x_1, x_2)-space of the effect of the stability contractor acting on the (x_1, x_2, x_3)-space

Definition 4. We define the *stability region* as

$$\mathbb{M} = \left\{ \mathbf{m} \,\middle|\, \exists \varepsilon > 0, \forall \mathbf{x}_0 \in [-\varepsilon, \varepsilon]^{\times n}, \lim_{k \to \infty} \|\mathbf{x}_k\| = 0 \right\} \tag{22}$$

To calculate an inner approximation of $\mathbb{M}$, we decompose the **m**-space into small boxes. Take one of these small boxes $[\mathbf{m}]$ and a box $[\mathbf{x}_0]$ around **0**. The box $[\mathbf{x}_0]$ should be small, but large compared to $[\mathbf{m}]$. If there exists $k > 0$ such that the system is stable, *i.e.* $[\mathbf{f}_c^k]([\mathbf{x}_0], [\mathbf{m}]) \subset [\mathbf{x}_0]$, then $[\mathbf{m}] \subset \mathbb{M}$.

Consider a situation where the landmarks are at positions $\mathbf{a} = (0, 0.1)^{\mathrm{T}}$, $\mathbf{b} = (0, -0.1)^{\mathrm{T}}$. To characterize the set $\mathbb{M}$, we build a paving of the plane with small boxes $[\mathbf{m}]$ of width 0.001. Taking $\varepsilon = \sqrt{w([\mathbf{m}])}$ and $[\mathbf{x}_0] = [-\varepsilon, \varepsilon]^{\times 2}$, we get the results depicted on Figure 7. The green area is proved to be inside $\mathbb{M}$. As a consequence, if the robot is located in the green area and if its initial localisation error is smaller than ε, then the classical Newton method will converge towards the actual position of the robot. On the other hand, we are not able to conclude anything about the stability of the localisation algorithm in the red area: this could require to take smaller boxes $[\mathbf{m}]$ in the **m**-space, a smaller initial condition $[\mathbf{x}_0]$, or this might also mean that the localisation system is not stable when initialised inside that specific choice of parameters.

6.4   Example 4 : Stability of a cyclic trajectory

This last example illustrates how our method can be applied to a real-life problem : a robot controlled by a state-machine is deployed in a lake; we will prove that the trajectory of the latter is stable.

6.4.1   Presentation of the problem

Let us consider a robot (*e.g.* an Autonomous Underwater Vehicle (AUV)) cruising in a closed area (*e.g.* a lake) at a constant speed. We consider that the robot is moving in a plane, for the sake of simplicity. The robot is controlled by an automaton and a heading controller. It also embeds a sensor (*e.g.* a sonar) to detect if the shore is close by. In this case, an event is triggered and the automaton changes its states :
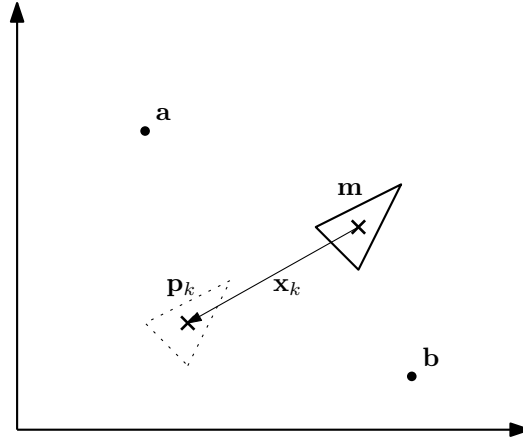
Figure 6: localisation problem formalisation

a new heading command is sent to the controller. Consider for instance the following sequence for the automaton:

1. Head East during 25 sec.

2. Head North until reaching the shore

3. Go South for 7.5 sec

4. Head West until reaching the shore

5. Go to 1.

6.4.2   Proving stability of the robot's trajectory

The goal is to prove that the robot's trajectory will converge towards a stable cycle.

Assume that the border is modelled by the function

$$x_2 = h(x_1) = 20\left(1 - \exp\left(-0.25x_1\right)\right) \tag{23}$$

Given the starting position of the robot $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$, the positions of the robot after each of the four transitions are respectively given by the following functions :

$$\mathbf{f}_1(\mathbf{x}) = (x_1 + 25v,\ x_2)^{\mathrm{T}}$$
$$\mathbf{f}_2(\mathbf{x}) = (x_1,\ h(x_1))^{\mathrm{T}}$$
$$\mathbf{f}_3(\mathbf{x}) = (x_1,\ x_2 - 7.5v)^{\mathrm{T}}$$
$$\mathbf{f}_4(\mathbf{x}) = \left(h^{-1}(x_2),\ x_2\right)^{\mathrm{T}}$$

where v denotes the cruising speed of the robot. Each of these functions can be modified to take model and sensor uncertainties into account, as we consider that the robot is cruising in dead-reckoning. Usually, these uncertainties grow with time if the robot does not perform exteroceptive measurements.

Let us define the cycle function $\mathbf{f}$, given by Equation (24):

$$\mathbf{f}(\mathbf{x}) \quad = \quad \mathbf{f}_4 \circ \mathbf{f}_3 \circ \mathbf{f}_2 \circ \mathbf{f}_1(\mathbf{x}) \tag{24}$$
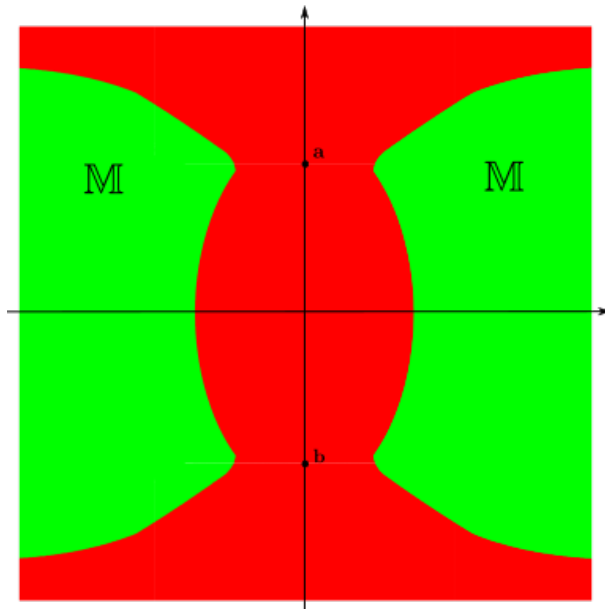
Figure 7: Stability region $\mathbb{M}$ of the localisation system

Proving stability of the cycle comes down to proving that the discrete-time system $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ is stable.

Figure 8 represents the evolution of $[\mathbf{x}]$ over steps of integration, as well as the intermediate steps. The initial box is $[\mathbf{x}_0] = ([1.5, 6.5], [9.5, 15.5])^{\mathrm{T}}$, and we model the robot getting lost by adding an uncertainty $[\mathbf{u}_i] = \frac{\|\mathbf{f}_i(\mathbf{x}) - \mathbf{x}\|}{v} \cdot [-\varepsilon, \varepsilon]^{\times 2}$ to each intermediate function $\mathbf{f}_i$, where $v = 1\,m/s$ is the cruising speed of the robot and $\varepsilon = 0.05\,m/s$ is the speed at which the robot is getting lost, *i.e.* the inflating rate of the robot's estimated position.

Since we have $[\mathbf{x}_1] \in [\mathbf{x}_0]$, we can conclude that the robot, equipped with the sensors named earlier and controlled by the given automaton, is going to perform a stable cycle in the lake.

7   Conclusion

In this paper, we have proposed a new approach to prove Lyapunov stability of a non-linear discrete-time system in a given neighbourhood $[\mathbf{x}_0]$, regardless of the system's uncertainties and the floating-point round-off errors. The principle is to perform a simulation based on the interval centred form, from an initial box $[\mathbf{x}_0]$ and until the current box $[\mathbf{x}_k]$ is strictly enclosed in $[\mathbf{x}_0]$. From the properties of the centred form we have proposed, we are able to conclude that a system is stable. The method applies to a large class of systems and can be used for arbitrary large dimensions, which is not the case for methods based on linearisation which are very sensitive to the dimension of the state vector. Additionally, contrary to the existing methods, ours is able to find a neighbourhood of radius δ inside which the system is stable.

The next step is to extend this approach to deal with continuous-time systems described by differential equations. This extension will require the introduction of interval integration methods [4, 22, 24, 30].

The interval community can be decomposed into two worlds :

- On the one hand, *small intervals* [19] are used in combination with linearisation methods and are able to propagate efficiently small uncertainties such as those related to round-off errors. They can solve high dimensional problems and do not require any bisections.
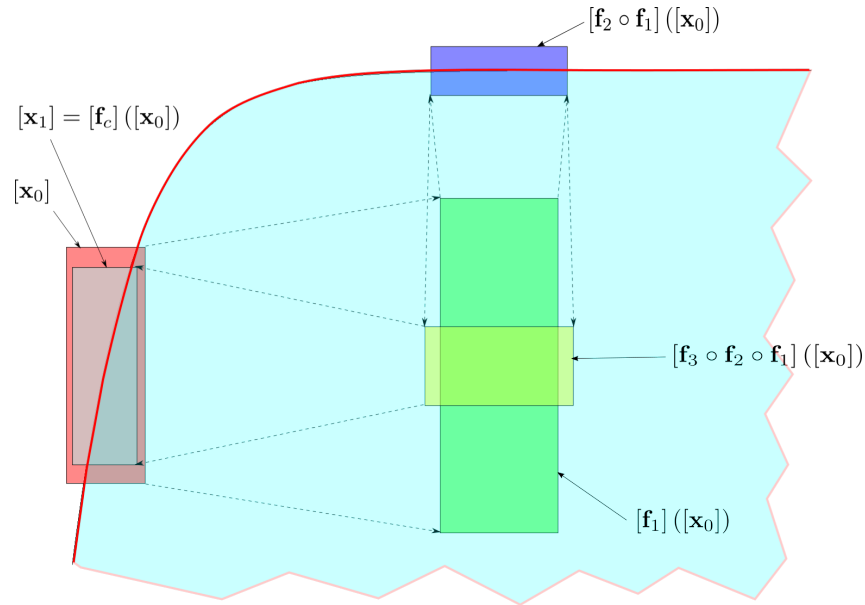
Figure 8: The robot rebounds on the border and follows a cyclic trajectory

- On the other hand, *large intervals* are mainly used for global optimization [8], solving equations [12], characterizing sets [10]. They used together with contractors techniques, constraint propagation and bisections algorithms.

This work belongs to the first world and makes use of small intervals. This is why the centred form is very efficient and why we do not perform any bisection. Now, in practice, the approaches developed by both communities can be used jointly : to solve real-life problems such as approximating basins of attraction, we will have to combine the two methodologies appropriately.

References

[1] M. V. A. Andrade, J. L. D. Comba & J. Stolfi (1994): *Affine Arithmetic*. In: *Interval'94*, St Petersburg.

[2] E. Asarin, T. Dang & A. Girard (2007): *Hybridization methods for the analysis of non-linear systems*. Acta Informatica 7(43), pp. 451–476, doi:10.1007/s00236-006-0035-7.

[3] G. Chabert & L. Jaulin (2009): *A Priori Error Analysis with Intervals*. SIAM Journal on Scientific Computing 31(3), pp. 2214–2230, doi:10.1137/070696982.

[4] A. Chapoutot, J. Alexandre Dit Sandretto & O. Mullier (2015): *Validated Explicit and Implicit Runge-Kutta Methods*. In: *Summer Workshop on Interval Methods*.

[5] C. Combastel (2005): *A State Bounding Observer for Uncertain Non-linear Continuous-time Systems based on Zonotopes*. In: *CDC-ECC '05*, doi:10.1109/CDC.2005.1583327.

[6] I. Fantoni & R. Lozano (2001): *Non-linear control for underactuated mechanical systems*. Springer-Verlag, doi:10.1007/978-1-4471-0177-2.

[7] G. Frehse (2008): *PHAVer: Algorithmic Verification of Hybrid Systems*. International Journal on Software Tools for Technology Transfer 10(3), pp. 23–48, doi:10.1007/978-3-540-31954-2_17.

[8] E. R. Hansen (1992): *Global Optimization using Interval Analysis*. Marcel Dekker, New York, NY.

[9] L. Jaulin & J. Burger (1999): *Proving stability of uncertain parametric models.* Automatica, pp. 627–632, doi:10.1016/S0005-1098(98)00201-5.

[10] L. Jaulin, M. Kieffer, O. Didrit & E. Walter (2001): *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics.* Springer-Verlag, London, doi:10.1007/978-1-4471-0249-6.

[11] Luc Jaulin & Fabrice Bars (2020): *Characterizing Sliding Surfaces of Cyber-Physical Systems.* Acta Cybernetica 24, pp. 431–448, doi:10.14232/actacyb.24.3.2020.9.

[12] R. B. Kearfott & V. Kreinovich, editors (1996): *Applications of Interval Computations.* Kluwer, Dordrecht, the Netherlands, doi:10.1007/978-1-4613-3440-8.

[13] M. Lhommeau, L Jaulin & L. Hardouin (2007): *Inner and outer approximation of capture basins using interval analysis.* ICINCO 2007.

[14] R. Lohner (1987): *Enclosing the solutions of ordinary initial and boundary value problems.* In E. Kaucher, U. Kulisch & Ch. Ullrich, editors: *Computer Arithmetic: Scientific Computation and Programming Languages,* BG Teubner, Stuttgart, Germany, pp. 255–286.

[15] T. Le Mézo, L. Jaulin & B. Zerr (2019): *Bracketing backward reach sets of a dynamical system.* In: *International Journal of Control,* doi:10.1080/00207179.2019.1643910.

[16] R. E. Moore (1966): *Interval Analysis.* Prentice-Hall, Englewood Cliffs, NJ.

[17] R. E. Moore (1979): *Methods and Applications of Interval Analysis.* SIAM, Philadelphia, PA, doi:10.1137/1.9781611970906.

[18] Ramon E Moore, R Baker Kearfott & Michael J Cloud (2009): *Introduction to interval analysis.* SIAM, doi:10.1137/1.9780898717716.

[19] A. Neumaier (1991): *Interval Methods for Systems of Equations.* Cambridge University Press, Cambridge, UK, doi:10.1017/CBO9780511526473.

[20] N. Ramdani & N. Nedialkov (2011): *Computing Reachable Sets for Uncertain Nonlinear Hybrid Systems using Interval Constraint Propagation Techniques.* Nonlinear Analysis: Hybrid Systems 5(2), pp. 149–162, doi:10.1016/j.nahs.2010.05.010.

[21] S. Ratschan & Z. She (2010): *Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-Like Functions.* SIAM Journal on Control and Optimization 48(7), pp. 4377–4394, doi:10.1137/090749955.

[22] N. Revol, K. Makino & M. Berz (2005): *Taylor models and floating-point arithmetic: proof that arithmetic operations are validated in COSY.* Journal of Logic and Algebraic Programming 64, pp. 135–154, doi:10.1016/j.jlap.2004.07.008.

[23] J. Rohn (1996): *An algorithm for checking stability of symmetric interval matrices.* IEEE Trans. Autom. Control 41(1), pp. 133–136, doi:10.1109/9.481618.

[24] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars & S. Veres (2019): *Reliable robot localization.* ISTE Group, doi:10.1002/9781119680970.

[25] S. M. Rump (2001): *INTLAB - INTerval LABoratory.* In J. Grabmeier, E. Kaltofen & V. Weispfennig, editors: *Handbook of Computer Algebra: Foundations, Applications, Systems,* Springer-Verlag, Heidelberg, Germany.

[26] P. Saint-Pierre (2002): *Hybrid kernels and capture basins for impulse constrained systems.* In C.J. Tomlin & M.R. Greenstreet, editors: *in Hybrid Systems: Computation and Control,* 2289, Springer-Verlag, pp. 378–392, doi:10.1007/3-540-45873-5_30.

[27] J.J. Slotine & W. Li (1991): *Applied nonlinear control.* Prentice Hall, Englewood Cliffs (N.J.). Available at http://opac.inria.fr/record=b1132812.

[28] W. Taha & A. Duracz: *Acumen: An Open-source Testbed for Cyber-Physical Systems Research.* In: *CYCLONE'15,* doi:10.1007/978-3-319-47063-4_11.

[29] Jian Wan (2007): *Computationally reliable approaches of contractive model predictive control for discrete-time systems*. PhD dissertation, Universitat de Girona, Girona, Spain.

[30] D. Wilczak & P. Zgliczynski (2011): *Cr-Lohner algorithm*. *Schedae Informaticae* 20, pp. 9–46, doi:10.4467/20838476SI.11.001.0287.