Guaranteed Characterization of Capture Basins of Nonlinear State-Space Systems

Nicolas Delanoue[1], Luc Jaulin[2], Laurent Hardouin[1] and Mehdi Lhommeau[1]

[1] Laboratoire d'Ingénierie des Systèmes Automatisés
Université d'Angers, 62 av. Notre Dame du Lac, 49000 Angers, France
{nicolas.delanoue, mehdi.lhommeau,
laurent.hardouin}@univ-angers.fr
[2] ENSIETA, 2 rue François Verny, 29806 Brest Cédex 09, France
luc.jaulin@ensieta.fr

Abstract. This paper proposes a new approach to solve the problem of computing the capture basin $\mathbb{C}$ of a target $\mathbb{T}$. The capture basin corresponds to the set of initial states such that the target is reached in finite time before possibly leaving of constrained set. We present an algorithm, based on interval analysis, able to characterize an inner and an outer approximation $\mathbb{C}^- \subset \mathbb{C} \subset \mathbb{C}^+$ of the capture basin. The resulting algorithm is illustrated on the Zermelo problem.

1 Introduction

The purpose of this paper is to present an algorithm based on guaranteed numerical computation which, given the dynamics of the system, provides an inner and outer approximation of the capture basin. We recall some definitions and notations related to capture basin. In the sequel, we consider nonlinear continuous-time dynamical systems of the form

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{x}(0) = \mathbf{x}_0, \end{cases} \tag{1}$$

were $\mathbf{x} \in \mathbb{R}^n$ is the state of the system with initial condition $\mathbf{x}_0$ at $t = 0$ and $\mathbf{u} \in \mathbb{R}^m$ is the control vector. We shall assume that the function $\mathbf{f}$ is sufficiently regular to guarantee that for all piecewise continuous function $\mathbf{u}(.)$ the solution of (1) is unique. The state vector $\mathbf{x}(t)$ is not allowed to exit a given compact set $\mathbb{K} \subset \mathbb{R}^n$ and the input $\mathbf{u}(t)$ should belong to a given compact set $\mathbb{U} \subset \mathbb{R}^m$.
We define the flow (see [1]) $\phi^t(\mathbf{x}_0, \mathbf{u})$ as the solution of (1) for the initial vector $\mathbf{x}_0$ and for the input function $\mathbf{u}(.)$. The path from t_1 to t_2 is defined by

$$\phi^{[t_1, t_2]}(\mathbf{x}_0, \mathbf{u}) \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^n \mid \exists t \in [t_1, t_2], \mathbf{x}(t) = \phi^t(\mathbf{x}_0, \mathbf{u}) \right\}. \tag{2}$$

Define a target set $\mathbb{T} \subset \mathbb{K} \subset \mathbb{R}^n$ as a closed set we would like to reach for one $t \geq 0$. The *capture basin* $\mathbb{C}$ of $\mathbb{T}$ is the set

$$\mathbb{C} \triangleq \big\{ \mathbf{x}_0 \in \mathbb{K} \mid \exists t \geq 0, \exists \mathbf{u}(.) \in \mathcal{F}([0, t] \to \mathbb{U}), \phi^t(\mathbf{x}_0, \mathbf{u}) \in \mathbb{T}$$
$$\text{and } \phi^{[0, t]}(\mathbf{x}_0, \mathbf{u}) \subset \mathbb{K} \big\}, \tag{3}$$

where $\mathcal{F}([0,t] \to \mathbb{U})$ represents the set of piecewise continuous functions from $[0,t] \to \mathbb{U}$. Then, $\mathbb{C}$ is the set of initial states $\mathbf{x} \in \mathbb{K}$ for which there exists an admissible control $\mathbf{u}$, and a finite time $t \geq 0$ such that the trajectory $\phi^{[0,t]}(\mathbf{x}_0, \mathbf{u})$ with the dynamic $\mathbf{f}$ under the control $\mathbf{u}$ lives in $\mathbb{K}$ and reaches $\mathbb{T}$ at time t.

The aim of the paper is to provide an algorithm able to compute an inner and an outer approximation of $\mathbb{C}$, i.e., to find two subsets $\mathbb{C}^-$ and $\mathbb{C}^+$ such that

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

Our contribution is twofold. First, in Section 2, we shall introduce interval analysis in the context of capture basin problem [2–4]. Second, in Section 3, we shall provide the first algorithm able to compute a garanteed inner and an outer approximation for capture basins. The efficiency of our approach will be illustrated on the Zermelo problem in Section 4. Section 5 will then conclude the paper.

2   Interval Analysis

The interval theory was born in the 60's aiming rigorous computations using finite precision computers (see [5]). Since its birth, it has been developed and it proposed today orignal algorithms for solving problems independently to the finite precision of computers computations, although reliable computations using finite precision remains one important advantage of the interval based algorithms [6].

An *interval* $[x]$ is a closed and connected subset of $\mathbb{R}$. A *box* $[\mathbf{x}]$ of $\mathbb{R}^n$ is a Cartesian product of n intervals. The set of all boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$. Note that $\mathbb{R}^n =]-\infty, \infty[\times \cdots \times]-\infty, \infty[$ is an element of $\mathbb{IR}^n$. Basic operations on real numbers or vectors can be extended to intervals in a natural way.

Example 1. If $[t] = [t_1, t_2]$ is an interval and $[\mathbf{x}] = \left[x_1^-, x_1^+\right] \times \left[x_2^-, x_2^+\right]$ is a box, then the product $[t] * [\mathbf{x}]$ is defined as follows

$$[t_1, t_2] * \left(\begin{bmatrix} x_1^-, x_1^+ \\ x_2^-, x_2^+ \end{bmatrix} \right) = \left(\begin{bmatrix} [t_1, t_2] * \left[x_1^-, x_1^+\right] \\ [t_1, t_2] * \left[x_2^-, x_2^+\right] \end{bmatrix} \right) =$$
$$\left(\begin{bmatrix} [\min(t_1 x_1^-, t_1 x_1^+, t_2 x_1^-, t_2\, x_1^+), \max(t_1 x_1^-, t_1 x_1^+, t_2 x_1^-, t_2 x_1^+)] \\ [\min(t_1 x_2^-, t_1 x_2^+, t_2 x_2^-, t_2 x_2^+), \max(t_1 x_2^-, t_1 x_2^+, t_2 x_2^-, t_2 x_2^+)] \end{bmatrix} \right).$$

2.1   Inclusion Function

The function $[\mathbf{f}](.) : \mathbb{IR}^n \to \mathbb{IR}^p$ is an *inclusion function* of a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^p$ if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) \triangleq \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \subset [\mathbf{f}]([\mathbf{x}]).$$
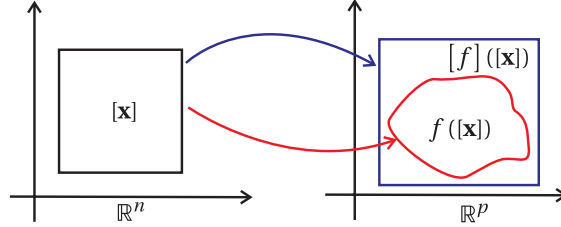
Fig. 1. Illustration of inclusion function.

Interval computation makes it possible to obtain inclusion functions of a large class of nonlinear functions, as illustrated by the following example.

Example 2. If $\mathbf{f}(x_1, x_2) \triangleq ((1 - 0.01x_2) x_1; (-1 + 0.02x_1) x_2)$, a methodology to obtain an enclosure of the image set $\mathbf{f}([10, 20], [40, 50])$ is as follows:

$$\mathbf{f}\begin{pmatrix}[40, 50]\\ [10, 20]\end{pmatrix} \subset \begin{pmatrix}(1 - 0.01 * [40, 50]) * [10, 20]\\ (-1 + 0.02 * [10, 20]) * [40, 50]\end{pmatrix} = \begin{pmatrix}(1 - [0.4, 0.5]) * [10, 20]\\ (-1 + [0.2, 0.4]) * [40, 50]\end{pmatrix} =$$

$$\begin{pmatrix}[0.5, 0.6] * [10, 20]\\ [-0.8, -0.6] * [40, 50]\end{pmatrix} = \begin{pmatrix}([5, 12])\\ ([-40, -24])\end{pmatrix}.$$

This methodology can easily be applied for any box $[x_1] \times [x_2]$ and the resulting algorithm corresponds to an inclusion function for $\mathbf{f}$.

The *interval union* $[\mathbf{x}] \sqcup [\mathbf{y}]$ of two boxes $[\mathbf{x}]$ and $[\mathbf{y}]$ is the smallest box which contains the union $[\mathbf{x}] \cup [\mathbf{y}]$. The *width* $w([\mathbf{x}])$ of a box $[\mathbf{x}]$ is the length of its largest side. The ε-*inflation* of a box $[\mathbf{x}] = [x_1^-, x_1^+] \times \cdots \times [x_n^-, x_n^+]$ is defined by

$$\text{inflate}([\mathbf{x}], \varepsilon) \quad \triangleq \quad [x_1^- - \varepsilon, x_1^+ + \varepsilon] \times \cdots \times [x_n^- - \varepsilon, x_n^+ + \varepsilon]. \quad (4)$$

2.2 Picard Theorem

Interval analysis for ordinary differential equations were introduced by Moore [5] (See [7] for a description and a bibliography on this topic). These methods provide numerically reliable enclosures of the exact solution of diffential equations. These techniques are based on Picard Theorem.

Theorem 1. *Let t_1 be a positive real number. Assume that $\mathbf{x}(0)$ is known to belong to the box $[\mathbf{x}](0)$. Assume that $\mathbf{u}(t) \in [\mathbf{u}]$ for all $t \in [0, t_1]$. Let $[\mathbf{w}]$ be a box (that is expected to enclose the path $\mathbf{x}(\tau), \tau \in [0, t_1]$). If*

$$[\mathbf{x}](0) + [0, t_1] * [\mathbf{f}]([\mathbf{w}], [\mathbf{u}]) \subset [\mathbf{w}], \quad (5)$$

where $[\mathbf{f}]([\mathbf{x}].[\mathbf{u}])$ is an inclusion function of $\mathbf{f}(\mathbf{x}, \mathbf{u})$, then, for all $t \in [0, t_1]$

$$\mathbf{x}([0, t_1]) \subset [\mathbf{x}](0) + [0, t_1] * [\mathbf{f}]([\mathbf{w}], [\mathbf{u}]). \quad (6)$$

2.3   Interval Flow

Definition: The *inclusion function of the flow* is a function

$$[\phi] : \begin{cases} \mathbb{IR} \times \mathbb{IR}^n \times \mathbb{IR}^m \to & \mathbb{IR}^n \\ ([t], [\mathbf{x}], [\mathbf{u}]) & \to [\phi]\,([t], [\mathbf{x}], [\mathbf{u}]) \end{cases}$$

such that

$$\forall t \in [t], \forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{u} \in \mathcal{F}([0, t] \to [\mathbf{u}]), \phi(t, \mathbf{x}, \mathbf{u}) \in [\phi]\,([t], [\mathbf{x}], [\mathbf{u}])$$

Using Theorem 1, one can build an algorithm computing an enclosure $[\mathbf{x}]([t])$ for the path $\mathbf{x}([t]) = \{\mathbf{x}(t), t \in [t]\}$ from an enclosure $[\mathbf{x}]$ for $\mathbf{x}(0)$. The principle of this algorithm is illustrated by Figure 2.

Algorithm 1: Inclusion function $[\phi]$.

Data: $[t] = [t_1, t_2], [\mathbf{x}](t_1), [\mathbf{u}]$
Result: $[\mathbf{x}](t_2), [\mathbf{w}]$

1 **begin**
2      $[\hat{\mathbf{x}}](t_2) := [\mathbf{x}](t_1) + (t_2 - t_1) * [\mathbf{f}]([\mathbf{x}](t_1), [\mathbf{u}]);$
3      $[\mathbf{v}] := [\mathbf{x}](t_1) \sqcup [\hat{\mathbf{x}}](t_2);$
4      $[\mathbf{w}] := \mathrm{inflate}([\mathbf{v}], \alpha.w([\mathbf{v}]) + \beta);$
5   **if** $[\mathbf{x}](t_1) + [0, t_2 - t_1] * [\mathbf{f}]([\mathbf{w}], [\mathbf{u}]) \not\subseteq [\mathbf{w}]$ **then**
6          $[\mathbf{x}](t_2) := \mathbb{R}^n$
7   return
8      $[\mathbf{x}](t_2) := [\mathbf{x}](t_1) + (t_2 - t_1) * [\mathbf{f}]([\mathbf{w}], [\mathbf{u}]);$
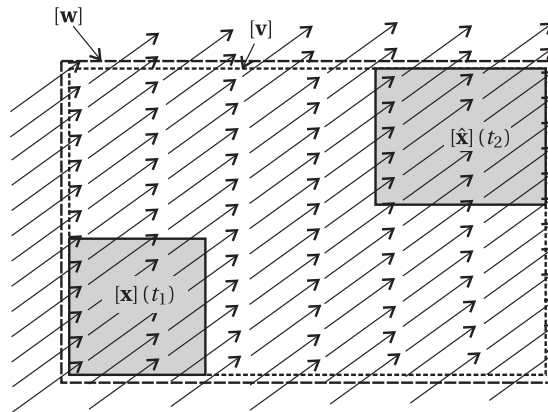9 **end**

Fig. 2. Principle of algorithm $[\phi]$.

Comments : The interval $[t] = [t_1, t_2]$ is such that $t_1 \geq 0$. Step 2 computes an estimation $[\hat{\mathbf{x}}](t_2)$ for the domain of all $\mathbf{x}(t_1)$ consistent with the fact that $\mathbf{x}(0) \in [\mathbf{x}]$. Note that, at this level, it is not certain that $[\hat{\mathbf{x}}](t_2)$ contains $\mathbf{x}(t_2)$. Step 3 computes the smallest box $[\mathbf{v}]$ containing $[\mathbf{x}](t_1)$ and $[\hat{\mathbf{x}}](t_2)$. At Step 4, $[\mathbf{v}]$ is inflated (see (4)) to provide a good candidate for $[\mathbf{w}]$. α and β are small positive numbers. Step 5 checks the condition of Theorem 1. If the condition is not satisfied, no bounds can be computed for $\mathbf{x}(t_2)$ and $\mathbb{R}^n$ is returned. Otherwise, Step 8 computes a box containing $\mathbf{x}(t_2)$ using theorem 1. ∎

The algorithm to we gave to compute the interval flow is very conservative. The pessimism can drastically be reduced by using the Lohner method [8].

3   Algorithm

This section presents an algorithm to compute an inner and an outer approximation of the capture basin. It is based on Theorem 2.

Theorem 2. *If $\mathbb{C}^-$ and $\mathbb{C}^+$ are such that $\mathbb{C}^- \subset \mathbb{C} \subset \mathbb{C}^+ \subset \mathbb{K}$, if $[\mathbf{x}]$ is a box and if $\mathbf{u} \in \mathcal{F}([0,t] \rightarrow \mathbb{U})$, then*

 (i) $[\mathbf{x}] \subset \mathbb{T} \Rightarrow [\mathbf{x}] \subset \mathbb{C}$
(ii) $[\mathbf{x}] \cap \mathbb{K} = \emptyset \Rightarrow [\mathbf{x}] \cap \mathbb{C} = \emptyset$
(iii) $\phi(t, [\mathbf{x}], \mathbf{u}) \subset \mathbb{C}^- \wedge \phi([0,t], [\mathbf{x}], \mathbf{u}) \subset \mathbb{K} \Rightarrow [\mathbf{x}] \subset \mathbb{C}$
(iv) $\phi(t, [\mathbf{x}], \mathbb{U}) \cap \mathbb{C}^+ = \emptyset \wedge \phi(t, [\mathbf{x}], \mathbb{U}) \cap \mathbb{T} = \emptyset \Rightarrow [\mathbf{x}] \cap \mathbb{C} = \emptyset$

Proof. (i) and (ii) are due to the inclusion $\mathbb{T} \subset \mathbb{C} \subset \mathbb{K}$. Since $\mathbb{T} \subset \mathbb{C}^- \subset \mathbb{C}$, (iii) is a consequence of the definition of the capture basin (see (3)). The proof of (iv) is easily obtained by considering (3) and in view of fact that $\mathbb{C} \subset \mathbb{C}^+ \subset \mathbb{K}$.

Finally, a simple but efficient bisection algorithm is then easily constructed. It is summarized in Algorithm 2. The algorithm computes both an inner and outer approximation of the capture basin $\mathbb{C}$. In what follows, we shall assume that the set $\mathbb{U}$ of feasible input vectors is a box $[\mathbf{u}]$. The box $[\mathbf{x}]$ to be given as an input argument for ENCLOSE should contain set $\mathbb{K}$.

Comments. Steps 4 and 7 uses Theorem 2, (i)-(iii) to inflate $\mathbb{C}^-$. Steps 5 and 8 uses Theorem 2, (ii)-(iv) to deflate $\mathbb{C}^+$.
where

   ε : ENCLOSE stops the bisecting procedure when the precision is reached ;
   $\mathbb{C}^-$: Subpaving (list of nonoverlapping boxes) representing an inner approximation of the capture basin, that is the boxes inside the capture basin $\mathbb{C}$;
   $\mathbb{C}^+$: Subpaving representing the outer approximation of the capture basin, that is the boxes outside $\mathbb{C}$ and the boxes for which no conclusion could be reached ;

These subpavings provide the following bracketing of the solution set :

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

Algorithm 2: ENCLOSE.

Data: $\mathbb{K}, \mathbb{T}, [\mathbf{x}], [\mathbf{u}]$
Result: $\mathbb{C}^-, \mathbb{C}^+$
begin

1    $\mathbb{C}^- \leftarrow \emptyset; \mathbb{C}^+ \leftarrow [\mathbf{x}]; \mathcal{L} \leftarrow \{[\mathbf{x}]\}$;

2   **while** $\mathcal{L} \neq \emptyset$ **do**

3   pop the largest box $[\mathbf{x}]$ from $\mathcal{L}$;

4   **if** $[\mathbf{x}] \subset \mathbb{T}$ **then**
       $\mathbb{C}^- \leftarrow \mathbb{C}^- \cup [\mathbf{x}]$;

5   **else if** $[\mathbf{x}] \cap \mathbb{K} = \emptyset$ **then**
       $\mathbb{C}^+ \leftarrow \mathbb{C}^+ \backslash [\mathbf{x}]$;

6   take $t \geq 0$ and $\mathbf{u} \in [\mathbf{u}]$

7   **if** $[\phi](t, [\mathbf{x}], \mathbf{u}) \subset \mathbb{C}^-$ and $[\phi]([0, t], [\mathbf{x}], \mathbf{u}) \subset \mathbb{K}$ **then**
       $\mathbb{C}^- \leftarrow \mathbb{C}^- \cup [\mathbf{x}]$;

8   **else if** $[\phi](t, [\mathbf{x}], [\mathbf{u}]) \cap \mathbb{C}^+ = \emptyset$ and $[\phi](t, [\mathbf{x}], [\mathbf{u}]) \cap \mathbb{T} = \emptyset$ **then**
       $\mathbb{C}^+ \leftarrow \mathbb{C}^+ \backslash [\mathbf{x}]$;

9   **else if** $w([\mathbf{x}]) \geq \varepsilon$ **then**
       bisect $[\mathbf{x}]$ and store the two resulting boxes into $\mathcal{L}$;

end

---

4 Experimentations

This section presents an application of Algorithm 2. The algorithm has been implemented in $C++$ using Profil/BIAS interval library and executed on a PentiumM 1.4Ghz processor. As an illustration of the algorithm we consider the Zermelo problem [9, 10]. In control theory, Zermelo has described the problem of a boat which wants to reach an island from the bank of a river with strong currents. The magnitude and direction of the currents are known as a function of position. Let $f(x_1, x_2)$ be the water current of the river at position (x_1, x_2). The method for computing the expression of the speed vector field of two dimensional flows can be found in [11]. In our example the dynamic is nonlinear,

$$f(x_1, x_2) \triangleq \left(1 + \frac{x_2^2 - x_1^2}{(x_1^2 + x_2^2)^2}, \; \frac{-2x_1 x_2}{(x_1^2 + x_2^2)^2}\right).$$

The speed vector field associated to the dynamic of the currents is represented on Figure 3.

Let $\mathbb{T} \triangleq \mathcal{B}(0, r)$ with $r = 1$ be the island and we set $\mathbb{K} = [-8, 8] \times [-4, 4]$, where $\mathbb{K}$ represents the river. The boat has his own dynamic. He can sail in any direction at a speed v. Figure 4 presents the two-dimensional boat. Then, the global dynamic is given by

$$\begin{cases} x_1'(t) = 1 + \dfrac{x_2^2 - x_1^2}{(x_1^2 + x_2^2)^2} + v\cos(\theta) \\ x_2'(t) = \dfrac{-2x_1 x_2}{(x_1^2 + x_2^2)^2} + v\sin(\theta) \end{cases},$$

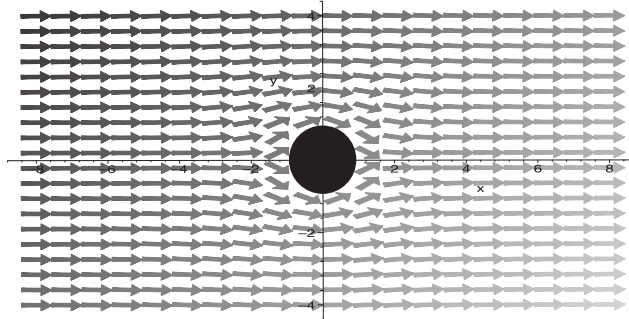where the controls $0 \leq v \leq 0.8$ and $\theta \in [-\pi, \pi]$.

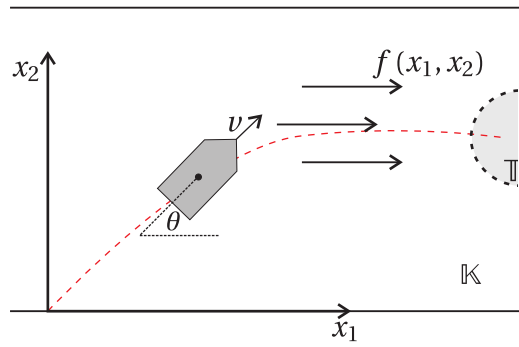**Fig. 3.** Vector field of the currents.
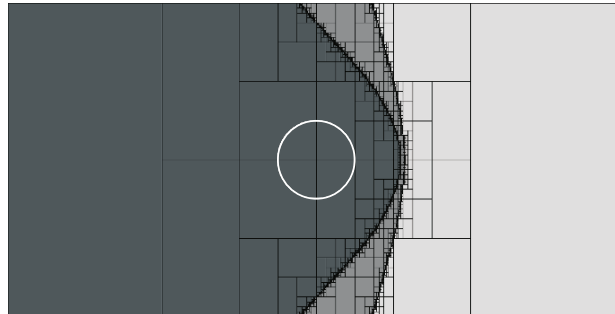

Fig. 4. Zermelo's problem.

Fig. 5. Two dimensional example of ENCLOSE algorithm.

Figure 5 shows the result of the ENCLOSE algorithm, where the circle delimits the border of the target $\mathbb{T}$. Then, $\mathbb{C}^-$ corresponds to the union of all dark grey boxes and $\mathbb{C}^+$ corresponds to the union of both grey and light grey boxes. Thus, we have the following inclusion relation :

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

5   Conclusions

In this paper, a new approach to deal with capture basin problems is presented. This approach uses interval analysis to compute an inner an outer approximation of the capture basin for a given target. To fill out this work, different perspectives appear. It could be interesting to tackle problems in significantly larger dimensions. The limitation is mainly due to the bisections involved in the interval algorithms that makes the complexity exponential with respect to the number of variables. Constraint propagation techniques [12] make it possible to push back this frontier and to deal with high dimensional problems (with more than 1000 variables for instance). In the future, we plan to combine our algorithm with graph theory and guaranteed numerical integration [7, 13] to compute a guaranteed control $\mathbf{u}$.

References

1. Hirsch, M. W., Smale, S.: Differential Equations, Dynamical Systems, and Linear Algebra. ap, San Diego (1974)
2. Aubin, J.: Viability theory. Birkhuser, Boston (1991)
3. Saint-Pierre, P.: Approximation of the viability kernel. Applied Mathematics & Optimization 29 (1994) 187-209
4. Cruck, E., Moitie, R., Seube, N.: Estimation of basins of attraction for uncertain systems with affine and lipschitz dynamics. Dynamics and Control 11(3) (2001) 211-227
5. Moore, R.E.: Interval Analysis. Prentice-Hall, Englewood Cliffs, NJ (1966)
6. Kearfott, R. B., Kreinovich, V., eds.: Applications of Interval Computations. Kluwer, Dordrecht, the Netherlands (1996)
7. Nedialkov, N. S., Jackson, K. R., Corliss, G. F.: Validated solutions of initial value problems for ordinary differential equations. Applied Mathematics and Computation 105 (1999) 21-68
8. Lohner, R.: Enclosing the solutions of ordinary initial and boundary value problems. In Kaucher, E., Kulisch, U., Ullrich, C., eds.:Computer Arithmetic: Scientific Computation and Programming Languages. BG Teubner, Stuttgart, Germany (1987) 255-86
9. Bryson, A.E., Ho, Y.C.: Applied optimal control: optimization, estimation, and control. Halsted Press (1975)
10. Cardaliaguet, P., Quincampoix, M., Saint-Pierre, P.: Optimal times for constrained nonlinear control problems without local controllability. Applied Mathematics and Optimization 36 (1997) 21-42
11. Batchelor, G.K.: An introduction to fluid dynamics. Cambridge university press(2000)
12. L. Jaulin, M. Kieffer, O. Didrit, E. Walter: Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics. Springer-Verlag, London (2001)
13. Delanoue, N.: Algorithmes numriques pour l'analyse topologique. PhD dissertation, Universit d'Angers, ISTIA, France (decembre 2006) Available at www.istia.univ-angers.fr/~delanoue/.