



Capture basin approximation using interval analysis

Mehdi Lhommeau, Luc Jaulin, Laurent Hardouin

► To cite this version:

Mehdi Lhommeau, Luc Jaulin, Laurent Hardouin. Capture basin approximation using interval analysis. *International Journal of Adaptive Control and Signal Processing*, 2011, 25 (3), pp.264-272. 10.1002/acs.1195 . hal-00593261

HAL Id: hal-00593261

<https://ensta-bretagne.hal.science/hal-00593261>

Submitted on 9 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Capture Basin Approximation using Interval Analysis

M. Lhommeau*, L. Jaulin and L. Hardouin

SUMMARY

This paper proposes a new approach for computing the capture basin $\mathbf{C}$ of a target $\mathbf{T}$. The capture basin corresponds to the set of initial state vectors such that the target could be reached in finite time via an appropriate control input, before possibly leaving the target. Whereas classical capture basin characterization do not provide any guarantee on the set of state vectors that belong to the capture basin, interval analysis and guaranteed numerical integration allow us to avoid any indetermination. We present an algorithm able to provide guaranteed approximation of the inner $\mathbf{C}^-$ and an the outer $\mathbf{C}^+$ of the capture basin, such that $\mathbf{C}^- \subseteq \mathbf{C} \subset \mathbf{C}^+$. In order to illustrate the principle and the efficiency of the approach, a testcase on the "car on the hill" problem is provided. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: Nonlinear systems, Viability theory, Capture basin, Interval Analysis, Guaranteed Numerical Integration

1. Introduction

This paper deals with the control of dynamical systems of the form $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$, $t \geq 0$, with state $\mathbf{x}(t) \in \mathbb{R}^n$ and control $\mathbf{u}(t) \in \mathbf{U}$, in the presence of state constraint $\mathbf{x}(t) \in \mathbf{K} \subseteq \mathbb{R}^n$. The set $\mathbf{K} \times \mathbf{U}$ describes the "acceptable configurations of the system". Such problems of dynamic control under constraints refer to viability theory [2] (see [1] for a survey). The main concepts of the viability theory are

- *viable state* : a state is called viable if there exists at least one control function for which the whole trajectory from this state remains in $\mathbf{K}$ indefinitely.
- *Viability kernel* : the set of all viable states is called the viability kernel.
- *Capture basin* : let $\mathbf{T}$ be a subset of $\mathbf{K}$. The subset of initial states $\mathbf{x} \in \mathbf{K}$ such that $\mathbf{T}$ can be reached in finite time before possibly leaving $\mathbf{K}$ by at least one trajectory $\mathbf{x}(\cdot)$ is called the capture basin and is denoted $\mathbf{C}$.

The characterization of the capture basin has a wide range of applications, for example, in process control [8, 20] and system biology [3, 6]. Thus the development of methods for

*Correspondence to: Laboratoire d'Ingénierie des Systèmes Automatisés, Université d'Angers, France

the characterization of the capture basin for dynamical systems is an important area in control engineering. In general the computation of the viability kernel is not an easy task. The development of computational tools to support the numerous viability theory methods is an ongoing effort. The viability kernel and capture basin are computed thanks to viability algorithms ([19], [18]). Saint-Pierre's algorithm computes for a given grid $\mathbf{X}_h$ a discrete viability kernel that converges to the viability kernel when the grid resolution h tends to 0. Also several interesting results have recently appeared on numerical methods to compute viability kernels and capture basins [10], [5], [7] and [9]. Most of the time these approaches provide approximations based on the numerical diffusion induced by the discretization scheme.

We have shown in earlier works how to provide an inner and outer approximation of the capture basin using interval analysis and guaranteed numerical integration. The major improvement presented in this paper is given by the main results detailed in Theorem 4.1 and Theorem 4.2. In particular, these results allow to give a new algorithm where the exploration of the search space is organized in such a way to ensure the algorithm convergence. The proposed method to approximate capture basins has several advantages over traditional numerical methods to find capture basins : (1) it is guarantee, whereas numerical methods give only an approximation of the capture basin. Indeed, in conventional methods, it is possible that some states found in the capture basin cannot reach the target $\mathbf{T}$. In other words, the set found for the capture basin corresponds to an over-approximation of the real capture basin set. This problem is avoided thanks to the interval analysis and the guaranteed numerical integration. Indeed, the result given by the algorithm in this paper is two sets $\mathbf{C}^-$ and $\mathbf{C}^+$ such that $\mathbf{C}^- \subseteq \mathbf{C} \subseteq \mathbf{C}^+$; (2) Numerical methods generally do not allow the designer to specify up front the class of control inputs with respect to which the capture basin is to be found. In our algorithm, it is possible to specify, exactly, the desired controls ; (3) Our algorithm used with a low accuracy may very quickly eliminate a large portion of the space, not included in the capture basin, to be explored even with a rather large number of parameters.

The paper is organized as follows. Section 2 recalls some results related to continuous dynamical system and capture basin results. The algorithm to be given in the sequel is based on interval analysis and guaranteed numerical integration of ODE. Then, the main validated techniques are described and referenced briefly in Section 3. In Section 4, we present the new algorithm of capture basin approximation, using interval analysis and in particular the guaranteed numerical integration. An application to a nonlinear dynamical system is provided in Section 5. Finally, we discuss the results and draw some perspectives.

2. Capture basin

We consider a control system, defined by the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ be the *state vector* of the system and $\mathbf{u}(t) \in \mathbf{U}$ be the control vector. We shall assume that the set $\mathbf{U} \subseteq \mathbb{R}^m$ of control values is compact, the function $\mathbf{f} : \mathbb{R}^n \times \mathbf{U} \rightarrow \mathbb{R}^n$ is locally Lipschitz on $\mathbb{R}^n$. Associated to the differential equation (1), we define the *flow map*

$$\varphi(t; \mathbf{x}_0, \mathbf{u}) = \mathbf{x}(t), \quad (2)$$

where $\mathbf{x}(t)$ denotes the solution to (1) with the initial condition $\mathbf{x}(0) = \mathbf{x}_0$ and the control function $\mathbf{u} \in \mathcal{U}$, where $\mathcal{U} = \{\mathbf{u} : [0, t] \rightarrow \mathbf{U} \mid t \geq 0 \text{ and } \mathbf{u} \text{ is piecewise continuous}\}$ denotes the set of admissible controls. Given $\mathbf{X}_0$ a set of possible initial values $\mathbf{x}_0$, the reachable set of system (1) at the time t is

$$\begin{aligned} \varphi(t; \mathbf{X}_0, \mathbf{U}) &= \{\varphi(t; \mathbf{x}_0, \mathbf{u}) \mid \varphi(0; \mathbf{x}_0, \mathbf{u}) = \mathbf{x}_0 \text{ and} \\ &\varphi : [0, t] \times \mathbf{X}_0 \times \mathbf{U} \rightarrow \mathbb{R}^n \text{ is a solution of (1) for some } \mathbf{u} \in \mathcal{U}\}. \end{aligned} \quad (3)$$

The path from t_1 to t_2 is defined by

$$\varphi([t_1, t_2]; \mathbf{X}_0, \mathbf{U}) = \{\mathbf{X} \subseteq \mathbb{R}^n \mid \exists t \in [t_1, t_2], \mathbf{X} = \varphi(t, \mathbf{X}_0, \mathbf{U})\}. \quad (4)$$

Let $\mathbf{K} \subseteq \mathbb{R}^n$ be a constraint set and $\mathbf{T}$ (the "target") be a compact set in $\mathbf{K}$. The *capture basin* $\mathbf{C}$ is the subset of initial states of $\mathbf{K}$ from which there exists at least one solution of (1) inside $\mathbf{K}$ reaching the target $\mathbf{T}$ in finite time t :

$$\mathbf{C} = \{\mathbf{x}_0 \mid \exists t \geq 0, \exists \mathbf{u} \in \mathcal{U}, \varphi(t; \mathbf{x}_0, \mathbf{u}) \in \mathbf{T} \text{ and } \varphi([0, t]; \mathbf{x}_0, \mathbf{u}) \subseteq \mathbf{K}\}. \quad (5)$$

3. Interval Analysis and Guaranteed Numerical Integration of ODE

Interval analysis for ordinary differential equations was introduced by Moore [15] (See [17] for a description and a bibliography on this topic). These methods provide numerically reliable enclosures of the exact solution of differential equations.

3.1. Interval and Interval vectors

Interval analysis usually considers only closed intervals. The set of these intervals is denoted by $\mathbb{IR}$. An interval is usually denoted using brackets. An element of an interval $[x]$ is denoted by x . An interval vector (boxes) $[\mathbf{x}]$ of $\mathbb{R}^n$ is a Cartesian product of n intervals. If $[\mathbf{x}] = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_n, \bar{x}_n]$ is a box, then its width is

$$w([\mathbf{x}]) = \max_{1 \leq i \leq n} w([x_i]), \quad (6)$$

where $w([x_i]) = \bar{x}_i - \underline{x}_i$. The set of all boxes of $\mathbb{R}^n$ is denoted by $\mathbb{IR}^n$.

3.2. Inclusion functions

The main concept of interval analysis is the extension of real functions to intervals, which is defined as follows.

Inclusion function Let $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a continuous real function, and $[\mathbf{f}] : \mathbb{IR}^n \rightarrow \mathbb{IR}^m$ be an inclusion function. Then $[\mathbf{f}]$ is an inclusion function of $\mathbf{f}$ if and only if for every $[\mathbf{x}] \in \mathbb{IR}^n$, $\{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \subseteq [\mathbf{f}]([\mathbf{x}])$.

Hence, an interval inclusion allows computing enclosures of the image of boxes by real functions. It now remains to show how to compute such inclusions. The first step is to compute formally the interval extension of elementary functions. For example, we define $[\underline{x}, \bar{x}] + [\underline{y}, \bar{y}] := [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$. Similar simple expressions are obtained for other functions

like $-, \times, \div, x^n, \sqrt{x}, \exp, \dots$. This process gives rise to the so-called *interval arithmetic* (see [11]).

Then, an interval inclusion for real functions compound of these elementary operations is simply obtained by changing the real operations to their interval counterparts. This interval inclusion is called the *natural extension*.

Inclusion flow Let $[t_1, t_2] \in \mathbb{IR}$ be a time interval (with $t_1 \leq t_2$), let $[\mathbf{u}] \in \mathbb{IR}^m$ be an interval vector of controls, and let $[\mathbf{x}_0] \in \mathbb{IR}^n$ be an interval vector of initial states. $[\varphi]$ is an inclusion flow for φ if it satisfies

- (i) $[\varphi]([t_1, t_2]; [\mathbf{x}_0], [\mathbf{u}]) \supseteq \varphi([t_1, t_2]; [\mathbf{x}_0], [\mathbf{u}]) = \{\varphi(t; \mathbf{x}_0, \mathbf{u}) \mid t \in [t_1, t_2], \mathbf{x}_0 \in [\mathbf{x}_0], \mathbf{u} \in [\mathbf{u}]\}.$
- (ii) If $w([t_1, t_2]) \rightarrow 0, w([\mathbf{u}]) \rightarrow 0, w([\mathbf{x}_0]) \rightarrow 0$ then $w([\varphi]([t_1, t_2], [\mathbf{x}_0], [\mathbf{u}])) \rightarrow 0.$

Remark Let $[t_1, t_2]$ be a degenerated interval (*i.e.* $t_1 = t_2$). In the sequel, we denote by the real number t the degenerated interval $[t_1, t_2]$.

Usually the computation of the inclusion flow is based on the Banach fixed-point theorem and the application of the Picard-Lindelöf operator (see [4, 17] for details). It is also possible to obtain a tighter enclosure for the inclusion function $[\varphi](t; [\mathbf{x}_0], [\mathbf{u}])$ based on higher order Taylor series expansions. The enclosures thus obtained are said validated which is in contrast with conventional numerical integration techniques which derive approximations with unknown global error and where the accumulation of both truncation and roundoff errors may cause the computed solution to deviate widely from the real one.

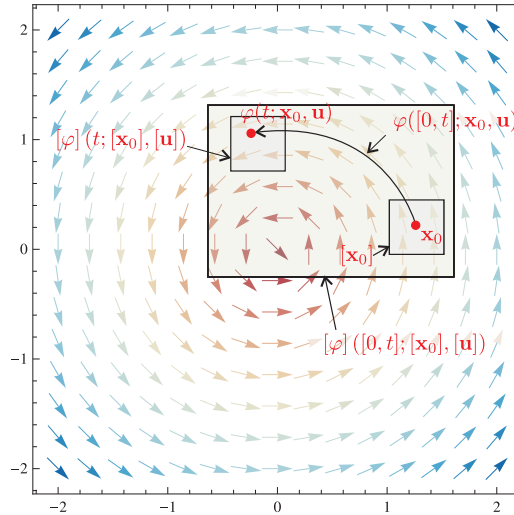


Figure 1. Illustration of the guaranteed enclosures of the exact solution of differential equation. The box $[\varphi]([0, t]; [\mathbf{x}_0], [\mathbf{u}])$ is an enclosure for the flow φ and the box $[\varphi](t; [\mathbf{x}_0], [\mathbf{u}])$ is an enclosure of the solution of the differential equation at time t

Given a bounded set $\mathbf{E}$ of complex shape, one usually defines an axis-aligned box or a paving, i.e. an union of non-overlapping boxes, $\mathbf{E}^+$ which contains the set $\mathbf{E}$: this is known as an outer approximation of it. Likewise, one also defines an inner approximation $\mathbf{E}^-$ which is contained in the set $\mathbf{E}$. Hence, we have the following property

$$\mathbf{E}^- \subseteq \mathbf{E} \subseteq \mathbf{E}^+. \quad (7)$$

4. Approximating the Capture Basin

This section presents an algorithm able to provide an inner and an outer approximation of the capture basin (see (5)). That is, we look for two sets, denoted $\mathbf{C}^-$ and $\mathbf{C}^+$, such that

$$\mathbf{C}^- \subseteq \mathbf{C} \subseteq \mathbf{C}^+.$$

Theorem 4.1. *Consider the control system (1) with $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$ be an initial condition. Let $\mathbf{T} \subseteq \mathbf{K} \subseteq \mathbb{R}^n$ be a target and let $\mathbf{C}$ be the capture basin of $\mathbf{T}$ in $\mathbf{K} \subseteq \mathbb{R}^n$. Then, the following statements hold*

- (i) $\mathbf{x}_0 \in \mathbf{T} \Rightarrow \mathbf{x}_0 \in \mathbf{C}$;
- (ii) $\mathbf{x}_0 \notin \mathbf{K} \Rightarrow \mathbf{x}_0 \notin \mathbf{C}$;
- (iii) $\exists t \geq 0, \exists \mathbf{u} \in \mathcal{U}, (\varphi(t; \mathbf{x}_0, \mathbf{u}) \in \mathbf{C} \wedge \varphi([0, t]; \mathbf{x}_0, \mathbf{u}) \subseteq \mathbf{K}) \Rightarrow \mathbf{x}_0 \in \mathbf{C}$;
- (iv) $\exists t \geq 0, \forall \mathbf{u} \in \mathcal{U}, (\varphi(t; \mathbf{x}_0, \mathbf{u}) \notin \mathbf{C} \wedge \varphi([0, t]; \mathbf{x}_0, \mathbf{u}) \cap \mathbf{T} = \emptyset) \Rightarrow \mathbf{x}_0 \notin \mathbf{C}$.

Proof

1. By definition of the capture basin (see (5)), $\mathbf{T} \subseteq \mathbf{C}$. This proves (i).
2. (ii) is immediate since $\mathbf{C} \subseteq \mathbf{K}$.
3. Suppose $\mathbf{x}_0 \notin \mathbf{C}$, then, from Equation (5) it follows

$$\forall t \geq 0, \forall \mathbf{u} \in \mathcal{U}, \varphi(t; \mathbf{x}_0, \mathbf{u}) \notin \mathbf{T} \vee \varphi([0, t]; \mathbf{x}_0, \mathbf{u}) \not\subseteq \mathbf{K}. \quad (8)$$

Then, from the definition of the capture basin given by equation (5) : if $\forall t \geq 0, \forall \mathbf{u} \in \mathcal{U}, \varphi(t; \mathbf{x}_0, \mathbf{u}) \notin \mathbf{T}$ then it means that it never reaches $\mathbf{C}$, otherwise it could reach $\mathbf{T}$, this implies that

$$\forall t \geq 0, \forall \mathbf{u} \in \mathcal{U}, \varphi(t; \mathbf{x}_0, \mathbf{u}) \notin \mathbf{T} \Rightarrow \varphi(t; \mathbf{x}_0, \mathbf{u}) \notin \mathbf{C}. \quad (9)$$

Thus, $\mathbf{x}_0 \notin \mathbf{C} \Rightarrow \forall t \geq 0, \forall \mathbf{u} \in \mathcal{U}, \varphi(t; \mathbf{x}_0, \mathbf{u}) \notin \mathbf{C} \vee \varphi([0, t]; \mathbf{x}_0, \mathbf{u}) \not\subseteq \mathbf{K}$ is true. Therefore the contrapositive (iii) is also true.

4. Assume $\mathbf{x}_0 \in \mathbf{C}$ and obtain a contradiction. Then, from Equation (5), we have

$$\mathbf{x}_0 \in \mathbf{C} \Rightarrow \exists \mathbf{u}_1 \in \mathcal{U}, \exists t_1 \geq 0, \varphi(t_1; \mathbf{x}_0, \mathbf{u}_1) \in \mathbf{T}. \quad (10)$$

By the assumption in (iv), there is a $t_2 \geq 0$ such that $\varphi(t_2; \mathbf{x}_0, \mathbf{u}) \notin \mathbf{C} = \emptyset$ and $\varphi([0, t_2]; \mathbf{x}_0, \mathbf{u}) \cap \mathbf{T} = \emptyset$ for all $\mathbf{u} \in \mathcal{U}$. We fix such a t_2 and set $\mathbf{u} = \mathbf{u}_1$, then $\varphi(t_2; \mathbf{x}_0, \mathbf{u}_1) \notin \mathbf{C} \wedge \varphi([0, t_2]; \mathbf{x}_0, \mathbf{u}_1) \cap \mathbf{T} = \emptyset$. Since, $\varphi(t_2; \mathbf{x}_0, \mathbf{u}_1) \notin \mathbf{C}$ and $\varphi(t_1; \mathbf{x}_0, \mathbf{u}_1) \in \mathbf{T}$ this implies, according to definition of the capture basin (see Equation (5)), $t_2 \geq t_1 \Rightarrow [0, t_1] \subseteq [0, t_2]$. Thus, $[0, t_1] \subseteq [0, t_2] \Rightarrow \varphi([0, t_1]; \mathbf{x}_0, \mathbf{u}_1) \subseteq \varphi([0, t_2]; \mathbf{x}_0, \mathbf{u})$. Clearly, since $\varphi([0, t_2]; \mathbf{x}_0, \mathbf{u}_1) \cap \mathbf{T} = \emptyset$, then $\varphi([0, t_1]; \mathbf{x}_0, \mathbf{u}_1) \cap \mathbf{T} = \emptyset$. This contradicts (10) and proves (iv).

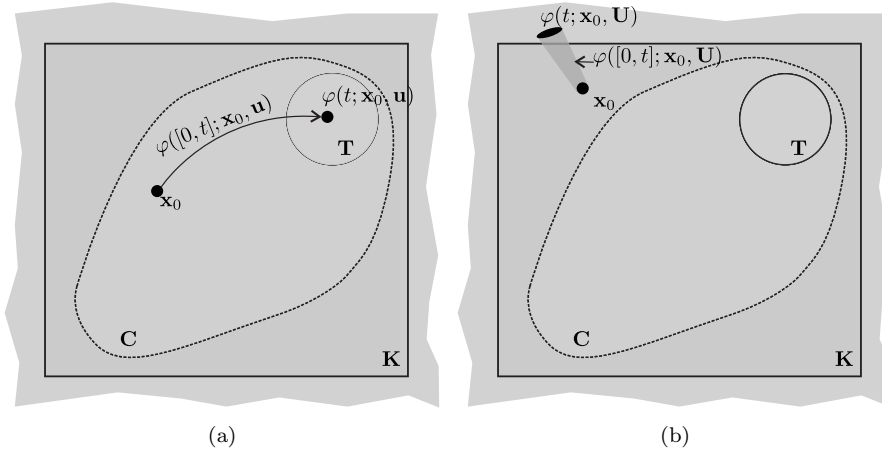


Figure 2. Illustration of the statements of Theorem 4.1. Figure (a) : implication (iii). Figure (b) : implication (iv).

In the sequel, we consider set-membership counterpart of Theorem 4.1. That is, the vector x_0 is replaced by a set X_0 . The interest to considering sets of values rather than punctual values is, on the one hand, to accelerate the algorithm presented in the following. On the other hand, to use the results of the interval analysis in order to obtain guaranteed results especially thanks to the guaranteed numerical integration.

Theorem 4.2. *Consider the control system (1). Let $T \subseteq K \subseteq \mathbb{R}^n$ be a target and let C be the capture basin of T in $K \subseteq \mathbb{R}^n$. Let $X_0 \subseteq \mathbb{R}^n$ be a set of initial conditions and let U be a set control values. Then, the following statements hold :*

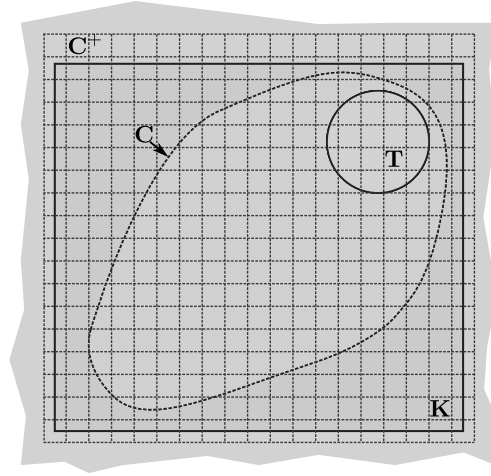
- (i) $X_0 \subseteq T \Rightarrow X_0 \subseteq C$
- (ii) $X_0 \cap K = \emptyset \Rightarrow X_0 \cap C = \emptyset$
- (iii) $\exists t \geq 0 \exists u \in U, (\varphi(t; X_0, u) \subseteq C \wedge \varphi([0, t]; X_0, u) \subseteq K) \Rightarrow X_0 \subseteq C$
- (iv) $\exists t \geq 0, (\varphi(t; X_0, U) \cap C = \emptyset \wedge \varphi([0, t]; X_0, U) \cap T = \emptyset) \Rightarrow X_0 \cap C = \emptyset$

Proof The proof is straightforward, starting from Theorem 4.1 : it suffices to consider the set-membership counterpart of the flow map (see (3)-(4)).

4.1. Algorithm

In order to discretize the continuous problem described in Theorem 4.2, we use a union of non-overlapping boxes in which is included the set K . This union of non-overlapping boxes corresponds to the set denoted C^+ (see Figure 3). Since, the sets considered in the algorithm are pavings boxes, it possible to apply the results of the guaranteed numerical integration of ODE (see Section 3).

The algorithm presented in this section proceeds in several steps, progressively defining the capture basin approximation.

Figure 3. Set C^+ is a union of non-overlapping boxes

The steps of the algorithm are the following.

```

input : Initialize the sets  $C^- = \emptyset$  and  $C^+$  is a union of boxes covering  $K$ 
output: Sets  $C^-$  and  $C^+$ , such that  $C^- \subseteq C \subseteq C^+$ 

1 foreach box  $[x_0]$  in  $C^+$  do
2   if  $[x_0] \subseteq T$  then
3      $C^- := C^- \cup [x_0]$ ;
4   else if  $[x_0] \cap K = \emptyset$  then
5      $C^+ := C^+ \setminus [x_0]$ 
6   else
7     take  $t \in \mathbb{R}^+$  and  $u \in [u]$  ;
8     if  $[\varphi](t; [x_0], u) \subseteq C^-$  and  $[\varphi]([0, t]; [x_0], u) \subseteq K$  then
9        $C^- := C^- \cup [x_0]$ 
10    else if  $[\varphi](t; [x_0], [u]) \cap C^+ = \emptyset$  and  $[\varphi]([0, t]; [x_0], [u]) \cap T = \emptyset$  then
11       $C^+ := C^+ \setminus [x_0]$ 
12    end
13  end
14 end

```

Algorithm 1: CAPTURE BASIN APPROXIMATION

Comments : The algorithm works as follows. Initially, the initial set C^+ corresponds to the union of boxes covering the set K , that is $K \subseteq C^+$. The principle of the algorithm is simple : for each box of C^+ it suffices to check the conditions of Theorem 4.2. If the conditions (i) (step 2 of the algorithm) or (iii) (step 5) hold, then the box is added to the set C^- . Note that, the condition (iii) can be carried out for several t and several control $u \in [u]$. Conversely, if the conditions (ii) (step 3) or (iv) (step 6) hold, this means that the box is not included in the

capture basin. Then, we remove the box from the set $\mathbf{C}^+$. Finally, the result of the algorithm is depicted on Figure 4. It is important to note that, all along the algorithm, we have the inclusion $\mathbf{C}^- \subseteq \mathbf{C} \subseteq \mathbf{C}^+$.

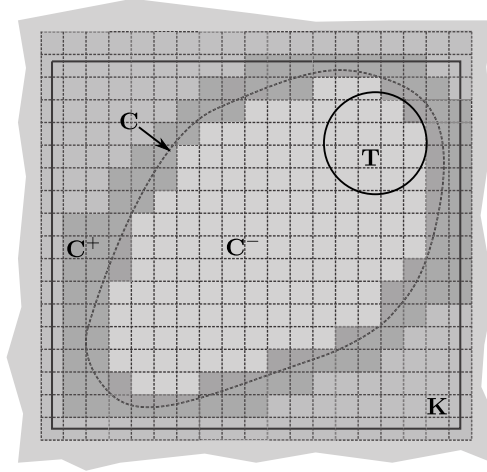


Figure 4. Final result of the Algorithm with $\mathbf{C}^- \subseteq \mathbf{C} \subseteq \mathbf{C}^+$

5. Illustration

To illustrate our approach, we have performed numerical experiments by considering the "car on the hill" problem [14],[16]. In this problem a car starts at the bottom of the hill and is supposed to reach the target on the other side of the hill. The difficulty is that gravity is stronger than the car's engine, and even at full throttle the car cannot accelerate up the steep slope. Then, to reach the target, the only solution is to first move away from the target and up to the opposite slope on the right. Then, by applying full throttle the car can build up enough inertia to carry it up the steep slope even though it is slowing down the whole way. We assume that the car stay on the hill ground. At a given time, the state of the car is determined by two scalar parameters : the position of the car on the hill (measured by the path length) s and its velocity $\dot{s}$ as shown in Figure 5. The control of the car motion is given by acting on the acceleration $\ddot{s}$. The shape of the hill is given by the function $g : s \rightarrow \theta, s \mapsto (1.1 \sin(1.2s) - 1.2 * \sin(1.1s))/2$. Calling $s = x_1$ and $\dot{s} = x_2$, we obtain a control system in standard form :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -9.81 \sin(g(x_1)) - \alpha x_2 + u \end{cases} \quad (11)$$

with the control $u \in [u] := [-2, 2]$ and the constraint $x_1 \in [-1, 13]$, $x_2 \in [-7, 7]$ and $\mathbf{K} = [0, 12] \times [-6, 6]$. Moreover, there is a damping (friction) force $\alpha = 0.7$ that is proportional to the velocity.

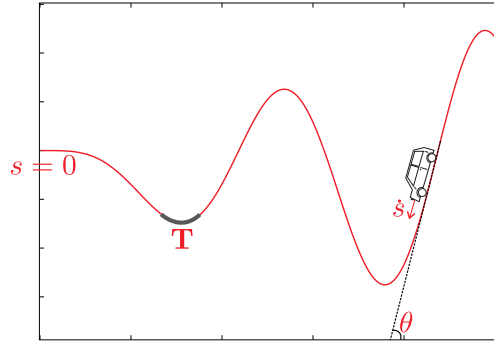


Figure 5. Car on the hill

In this illustration, the goal is to characterize the capture basin $\mathbf{C}$ of the target $\mathbf{T}$ in $\mathbf{K}$. The target corresponds to a bottom of the hill with a small positive or negative velocity. More precisely, we are looking for all the initial states $(x_1, x_2)^T$ in $\mathbf{K}$ from which there exists at least one solution of (11) inside $\mathbf{K}$ reaching the target $\mathbf{T} = [3.5, 4, 5] \times [-1, 1]$ in finite time. The final characterization of the capture basin is presented Figure 6. We observe two kinds of pavings : the paving $\mathbf{C}^-$ that corresponds to the inner approximation of the capture basin and paving $\mathbf{C}^+$ which is the outer approximation of the capture basin. In this example, to increase the accuracy, we have implemented adaptive size paving.

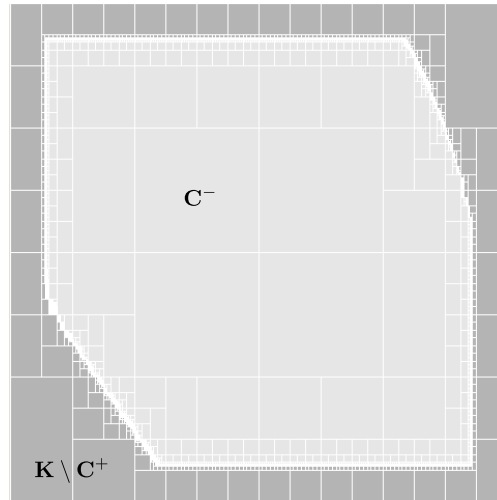


Figure 6. Representation of the capture basin of the "car on the hill" problem. The light grey zone, $\mathbf{C}^-$, is proved to be included in $\mathbf{C}$; the dark grey zone corresponds to $\mathbf{K} \setminus \mathbf{C}^+$.

The C++ code of the algorithm as well as movies illustrating our numerical experiments can be downloaded at <http://www.istia.univ-angers.fr/~lhommeau/capturebasin.html>.

6. Conclusion

In this paper, we have introduced a new interval-based method to characterize an inner and outer approximation of the capture basin. Our method differs from the other approaches, because we have no special assumption on the dynamical system. Moreover our algorithm provides guaranteed results. A testcase illustrating the efficiency of the approach has been treated on the "car on the hill" problem. In the near future, we are planning to extend this approach to other viability problems, such as computing the viability kernel.

REFERENCES

1. J.P. Aubin. *A survey of Viability Theory*, *SIAM Journal on Control and Optimization*, 28(2):749–788, 1990.
2. J.P. Aubin. *Viability theory*. Birkhäuser, Boston, 1991.
3. C. Bene, L. Doyen, and D. Gabay. A viability analysis for a bio-economic model. *Ecological Economics*, 36(3):385–396, March 2001.
4. M. Berz, and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4):361–369, 1998.
5. O. Bokanowski, S. Martin, R. Munos, and H. Zidani. An anti-diffusive scheme for viability problems. *Appl. Numer. Math.*, 56(9):1147–1162, 2006.
6. N. Bonneuil. Making ecosystem models viable. *Bulletin of Mathematical Biology*, 65(6):1081–1094, 2003.
7. N. Bonneuil. Computing the viability kernel in large state dimension. *Journal of Mathematical Analysis and Applications*, 323(2):1444–1454, 2006.
8. P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Optimal times for constrained nonlinear control problems without local controllability. *Applied Mathematics and Optimization*, 36:21–42, 1997.
9. P.-A. Coquelin, S. Martin, and R. Munos. A dynamic programming approach to viability problems. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, Honolulu, HI, 2007.
10. G. Deffuant, L. Chapel, and S. Martin. Approximating viability kernels with support vector machines. *IEEE Transactions on Automatic Control*, 52(5):933–937, 2007.
11. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.
12. M. Lhommeau, L. Jaulin and L. Hardouin. Inner and outer approximation of capture basins using interval analysis. In *4th International Conference on Informatics in Control, Automation and Robotics (ICINCO'07)*, Angers, France, 2007.
13. N. Delanoue, L. Jaulin, L. Hardouin, and M. Lhommeau. Guaranteed characterization of capture basins of nonlinear state-space systems. *Lecture Notes in Electrical Engineering*. Springer, 2008.
14. Andrew Moore. Variable resolution dynamic programming: Efficiently learning action maps in multivariate real-valued state-spaces. In L. Birnbaum and G. Collins, editors, *Machine Learning: Proceedings of the Eighth International Conference*, 340 Pine Street, 6th Fl., San Francisco, CA 94104, June 1991. Morgan Kaufmann.
15. R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
16. Rémi Munos and Andrew Moore. Variable resolution discretization in optimal control. *Mach. Learn.*, 49(2-3):291–323, 2002.
17. N.-S. Nedialkov, K.-R. Jackson, and G.-F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105:21–68, 1999.
18. D. Pujal and P. Saint-Pierre. Capture basin algorithm for evaluating and managing complex financial instruments. (186), 2006.
19. P. Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics & Optimization*, 29:187–209, 1994.
20. R.-J. Spiteri, D.-K. Pai, and U.-M. Ascher. Programming and control of robots by means of differentialalgebraic inequalities. *IEEE Transactions on Robotics and Automation*, 16(2):135–145, 2000.