Approximating Backstop Viability Kernels

Guillaume Deffuant, Isabelle Alvarez, Sophie Martin, Patrick Saint Pierre

APPROXIMATING BACKSTOP VIABILITY KERNELS

GUILLAUME DEFFUANT , ISABELLE ALVAREZ , SOPHIE MARTIN , AND PATRICK SAINT-PIERRE

Abstract. This paper defines a specific viability domain called backstop viability kernel and proposes a new algorithm that approximates such sets in a compact constraint set of d dimensions. The algorithm uses a set approximation technique based on a sample chosen in a regular grid of n^d vertices covering the considered constraint set. We show that, in some conditions, the result of the algorithm is the direct approximation of the backstop viability kernel by this approximation technique. This theoretical result is illustrated on examples for which the viability kernel is equal to the backstop viability kernel and can be derived analytically, using two set approximation methods: the nearest neighbour, ensuring a convergence rate in $\mathcal{O}(n^{-1})$, and a recently developed technique, the recursive simplex stars (resistars), ensuring a convergence rate in $\mathcal{O}(n^{-2})$.

Key words. viability kernel, viability domain, resistar.

1. Introduction.

Viability theory [2, 4] addresses the problem of maintaining a controlled dynamical system inside a given set of states, generally called the constraint set. This framework is particularly relevant for modelling sustainability problems in which the constraint set is interpreted as an acceptable or desirable property of the system that should be sustained [14, 9, 27, 22, 24]. It appeared also relevant in a variety of engineering problems [30, 18, 23] and in finance and economics [15, 29]. In [21, 8, 13, 26] viability theory is at the core of a mathematical definition of resilience, viewed as the capacity of the system to viably restore the property if it has been lost. Recently, [17] extended this view to a general theory of sustainable management.

One of the main concepts of viability theory is the viability kernel, the set of states from which the system can remain indefinitely in the constraint set. From a state located outside the viability kernel, it is certain that the dynamical system will cross the limits of the constraint set after a finite time, whatever the chosen controls over time. The viability kernel is also important because it is easy to derive from it a variety of control policies keeping the system indefinitely inside the constraint set (and actually also inside the viability kernel itself).

Generally, it is not possible to determine a viability kernel analytically and several methods provide numerical approximations [5, 6, 10, 12, 9, 14, 19, 20, 1]. In the frequent case of dynamics defined with ordinary differential equations in a continuous state space, most algorithms start from a regularly distributed sample of points (vertices of a regular grid) covering the constraint set and another one covering the control space. The first algorithm proposed by Saint-Pierre [28] uses a discrete approximation of the dynamical system on the vertices of a regular mesh. The algorithm computes a sequence of mesh subsets, until reaching a fixed point. The final set provides a discrete approximation of the viability kernel, which converges to the viability kernel when the time step tends to 0 and n, the number of points by grid axis, tends to infinity.

In the particular case of viability kernels defined as epigraphs of a function, [7] showed that the approximation error of the Saint-Pierre algorithm is in $\mathcal{O}(n^{-1})$, when choosing a time step in $\mathcal{O}(n^{-1})$. In the general case, [25] showed that the approximation error (defined with the Hausdorff distance) of the Saint-Pierre algorithm is linear in n^{-1} and in the time step, if the problem satisfies some specific conditions (the shadowing property in particular). As far as we know, there is no other algorithm for which the convergence rate has been established, even in restrictive conditions.

In this paper, we focus on approximating the backstop viability kernel, a subset of the viability kernel, with the aim to get a better convergence rate.

The first contribution of this paper is theoretical. It defines the backstop viability kernel as the union of all viability domains satisfying a specific property. It introduces the extended discrete time in which the control can change at each time step (as in usual discrete time dynamics) and also when the system is about to leave the constraint set K, and extends the definition of the backstop viability kernel to the extended discrete time. It establishes that, in some conditions, the backstop viability kernels in extended discrete time and in continuous time are equal.

The second contribution is an algorithm approximating backstop viability kernels. Approximating backstop viability kernels is easier than approximating viability kernels in general, because it boils down to testing long trajectories that change control at most once, starting from points located in a limited subset of the state space. The algorithm uses a set approximation technique which is assumed based on a sample derived from a regular grid of n^d points covering K. In some conditions, the final result is equal to the direct approximation of the exact backstop viability kernel by the set approximation technique. Therefore, the Hausdorff distance between the backstop viability kernel and its approximation by our algorithm when n grows is the same as if the set approximation technique was directly applied to the backstop viability kernel itself.

The third contribution is a report on tests of the algorithm on viability problems for which the backstop viability kernel is equal to the viability kernel and can be derived analytically. The tests use two approximation techniques: the nearest neighbour and a recently developed method, the resistar surfaces [11]. These methods guarantee (if the backstop viability kernel holds some smoothness properties) a Hausdorff approximation error respectively in $\mathcal{O}(n^{-1})$ and $\mathcal{O}(n^{-2})$. The results of the tests are in line with the theory.

The remaining of the paper is organised as follows: Section 2 defines the backstop viability kernel in continuous time and extended discrete time control and determines some conditions in which these sets are equal. Section 3 presents the approximation algorithm and the theoretical study of its convergence. Section 4 reports tests of the algorithm convergence on examples. The final section discusses the contributions of the paper.

2. Backstop viability kernels.

2.1. Continuous time.

2.1.1. Viability problem and viability kernel. We consider a controlled dynamical system defined by its state $x(t) \in \mathbb{R}^d$ which can be influenced by a control $u(t)$, chosen in a compact set $U \subset \mathbb{R}^m$, defined by an ordinary differential equation where φ is a continuous function from $\mathbb{R}^d \times \mathbb{R}^m$ to $\mathbb{R}^d$:

$$\left\{ \begin{array}{l} \dot{x}(t) = \varphi(x(t), u(t)) \\ u(t) \in U. \end{array} \right. \tag{2.1}$$

A viability problem from point x_0 is to determine a measurable function $u(t) : \mathbb{R}_+ \to U$, such that the trajectory from x_0 when applying $u(.)$ remains for all t in compact set $K \subset \mathbb{R}^n$.

Integrating equation 2.1, from initial state x_0 and for a chosen control function $u(.)$, determines the successor of x_0 at time t as follows:

$$x_{x_0,u(.)}(t) := x_0 + \int_0^t \varphi(x_{x_0,u(.)}(t'), u(t'))dt'. \tag{2.2}$$

We also denote all the successors (or the trajectory) of x_0 during time interval $[0,t]$ when applying control function $u(.)$ as follows:

$$X_{x_0,u(.)}(t) := \bigcup_{t' \in [0,t]} x_{x_0,u(.)}(t'). \tag{2.3}$$

DEFINITION 2.1. *The viability kernel $Viab(K)$ of set K under the dynamics defined by function φ and control set U is the set of states x_0, for which there exists a control function $u(.)$ such that all successors of x_0 when applying $u(.)$ are in K:*

$$Viab(K) = \left\{x_0 \in K, \exists u(.) : \mathbb{R}_+ \to U, \forall t \in \mathbb{R}_+, x_{x_0,u(.)}(t) \in K\right\}, \tag{2.4}$$

where $\mathbb{R}_+$ denotes the set of positive real numbers.

Determining the viability kernel is important because a variety of control functions $u(.)$ keeping the trajectory indefinitely within K can easily be derived from it. However, the analytical determination of this set is generally impossible and it should therefore be approximated numerically. In this paper, we define the backstop viability kernel, a subset of the viability kernel which is easier to approximate.

2.1.2. Backstop viability domain and backstop viability kernel. We need a few preliminary notations and definitions:

- For $(x,y) \in K^2$, $D(x,y)$ denotes the distance between x and y in K, which is the length of the shortest continuous path connecting x to y in K (K is supposed to be path-connected). Of course, if K is convex, $D(x,y) = \|x-y\|$, the usual Euclidean distance.
- For a set $A \subset K$, the distance from $x \in K$ to A in K is: $D(x,A) = \inf_{y \in A} D(x,y)$. By convention: $D(x,\emptyset) = \infty$.
- For any two sets (A,B) included in K, the distance from A to B in K is: $D(A,B) = \inf_{x \in A} D(x,B)$.
- $\mathbb{R}_+^*$ denotes the set of strictly positive real numbers.
- For a couple of sets A and B, such that $B \subset A$, $A - B$ denotes the complementary set of B in A.
- If $u(t') = u$ for all $t' \in [0,t]$, then $x_{x_0,u(.)}(t)$ and $X_{x_0,u(.)}(t)$ are respectively denoted $x_{x_0,u}(t)$ and $X_{x_0,u}(t)$.

DEFINITION 2.2. *The exit time from a set A for an evolution starting from x_0 when applying the control function $u(.)$ is:*

$$\tau_A^\sharp(x_0, u(.)) = \inf\{t \in \mathbb{R}_+, x_{x_0,u(.)}(t) \notin A\}. \tag{2.5}$$

The exit time from set A and point x_0 obtained when applying a constant control u is denoted: $\tau_A^\sharp(x_0,u)$.

DEFINITION 2.3. *For any couple $(u,v) \in U^2$, the constant control from $x_0 \in K$ with one possible bounce on the boundary of K, denoted $(u,v)_{x_0,K}(.)$, is the function*

from $\mathbb{R}_+$ to U defined as follows:

$$(u,v)_{x_0,K}(t) = \begin{cases} u \text{ for } 0 \leq t \leq \tau_K^\sharp(x_0,u) \\ v \text{ for } t > \tau_K^\sharp(x_0,u). \end{cases} \tag{2.6}$$

The point reached at time t when applying this control function from x_0 is denoted $x_{x_0,(u,v)_K}(t)$ and the corresponding trajectory is denoted $X_{x_0,(u,v)_K}(t)$.

We can now define a backstop viability domain.

DEFINITION 2.4. *The set $V \subset K$ is a backstop viability domain of K under dynamics 2.1 for time threshold $\theta \in \mathbb{R}_+^*$ iff there exists a set $B \subset V$ such that:*

- *$D(B, K-V) > 0$;*
- *For all $x_0 \in B$, there exist $(u,v) \in U^2$ such that $X_{x_0,(u,v)}(\theta) \subset V$;*
- *For all $x_0 \in V - B$, there exists $(u,v) \in U^2$, such that:*
  - *either for all $t \in \mathbb{R}_+, x_{x_0,(u,v)_K}(t) \in V$,*
  - *or there exists $t \in \mathbb{R}_+^*$, $t \geq \theta \mid X_{x_0,(u,v)_K}(t) \subset V$ and $x_{x_0,(u,v)_K}(t) \in B$.*

B is called a backstop in V for time threshold θ, $V - B$ is called the catch of B in V, $D(B, K-V)$ the width of the catch of B in V. By convention, the empty set is a backstop viability domain for any catch width and any time threshold.

PROPOSITION 2.5. *Let V be a backstop viability domain of K and B a backstop in V of catch width $\delta \in \mathbb{R}_+^*$ for time threshold $\theta \in \mathbb{R}_+^*$. Let $B' \subset V$ be such that $B \subset B'$ and $D(B', K-V) = \delta'$ with $\delta' \leq \delta$. For any $\theta \in \mathbb{R}_+^*$ such that $0 < \theta' \leq \theta$, B' is a backstop in V for time threshold θ' of catch width δ'.*

Proof. Consider $x_0 \in V - B'$. Because $B \subset B'$, $x_0 \in V - B$. Therefore, there exist $(u,v) \in U^2$ such that:

- either $\forall t \in \mathbb{R}+, x_{x_0,(u,v)_K}(t) \in V$,
- or $\exists t \in \mathbb{R}_+^*$, $t \geq \theta$ such that $X_{x_0,(u,v)_K}(t) \subset V$ and $x_{x_0,(u,v)_K}(t) \in B$. Therefore $t \geq \theta'$ and since $B \subset B'$, $x_{x_0,(u,v)_K}(t) \in B'$.

Now consider $x_0 \in B'$. If $x_0 \in B$, there exists $(u,v) \in U^2$ such that $X_{x_0,u}(\theta) \subset V$ thus $X_{x_0,u}(\theta') \subset V$. If $x_0 \in B' - B$, then $x_0 \in V - B$ and there exists $t \in \mathbb{R}_+^*$, $t \geq \theta$ such that $X_{x_0,(u,v)_K}(t) \subset V$, and $t \geq \theta$ implies $t \geq \theta'$.

Therefore, B' satisfies the conditions for being a backstop in V for time threshold θ' and its catch width is δ' by definition. ∎

PROPOSITION 2.6. *For any $(\delta, \theta) \in (\mathbb{R}_+^*)^2$, the union of all backstop viability domains of K under dynamics 2.1 including a backstop of catch width greater or equal to δ for time threshold θ is a backstop viability domain of K including a backstop of catch width greater or equal to δ for time threshold θ. It is called the backstop viability kernel of catch width δ for time threshold θ and denoted $BSViab(\delta, \theta, K)$.*

Proof. Let $\mathcal{D}_{Bstop}(\delta, \theta, K)$ be the set of all backstop viability domains in K including a backstop of catch width at least δ for time threshold θ. Let:

$$V = \bigcup_{V' \in \mathcal{D}_{Bstop}(\delta,\theta,K)} V' \text{ and } B = \bigcup_{V' \in \mathcal{D}_{Bstop}(\delta,\theta,K)} B_{V'}, \tag{2.7}$$

where $B_{V'}$ is a backstop of V' of catch width δ for time threshold θ, for each $V' \in \mathcal{D}_{Bstop}(\delta, \theta, K)$.

Consider $x_0 \in B$. There exists $V' \in \mathcal{D}_{Bstop}(\delta, \theta, K)$ such that $x_0 \in B_{V'}$, and $D(x_0, K - V') \geq \delta$. Because $(K - V) \subset (K - V')$, $D(x_0, K - V) \geq \delta$.

Consider $x_0 \in V - B$. There exists $V' \in \mathcal{D}_{Bstop}(\delta, \theta, K)$ such that $x_0 \in V'$, and for all $V'' \in \mathcal{D}_{Bstop}(\delta, \theta, K)$, $x_0 \notin B_{V''}$, which implies $x_0 \in V' - B_{V'}$. Therefore, there exist $(u, v) \in U^2$ such that:

- either $\forall t \in \mathbb{R}_+, x_{x_0,(u,v)_K}(t) \in V'$, then $\forall t \in \mathbb{R}_+, x_{x_0,(u,v)_K}(t) \in V$,
- or $\exists t \in \mathbb{R}_+^*$, $t \geq \theta$ such that $X_{x_0,(u,v)_K}(t) \subset V'$ then $X_{x_0,(u,v)_K}(t) \subset V$ because $V' \subset V$ and $x_{x_0,(u,v)_K}(t) \in B_{V'}$, then $x_{x_0,(u,v)_K}(t) \in B$, because $B_{V'} \subset B$.

Consider $x_0 \in B$. There exists $V' \in \mathcal{D}_{Bstop}(\delta, \theta, K)$ such that $x_0 \in B_{V'}$. Therefore, there exists $(u, v) \in U^2$ such that $X_{x_0,(u,v)_K}(\theta) \subset V'$, hence $X_{x_0,(u,v)_K}(\theta) \subset V$. ∎

DEFINITION 2.7. *If there exists* $(\delta_0, \theta_0) \in (\mathbb{R}_+^*)^2$ *such that:*

$$BSViab(\delta_0, \theta_0, K) = \bigcup_{(\delta,\theta)\in(\mathbb{R}_+^*)^2} BSViab(\delta, \theta, K), \tag{2.8}$$

then this set is called the backstop viability kernel of K *under dynamics 2.1 and denoted* $BSViab(K)$.

Section 4.1 describes examples of viability problems for which the backstop viability kernel exists.

2.2. Extended discrete time.

2.2.1. Viability kernel in extended discrete time. We now define the control functions in extended discrete time (abbreviated as EDT), namely the control function for which it is possible to change the control at each clock tick (as usual in discrete time) and also just before the system leaves the constraint set K. This extended discrete time will be used in the algorithm approximating backstop viability kernels.

DEFINITION 2.8. *The EDT control function set of time step* $\theta \in \mathbb{R}_+^*$*, for constraint set* K*, dynamics 2.1 and starting from* $x_0 \in K$ *is denoted* $\mathcal{E}_\theta(x_0, K, U)$ *and is the set of piece-wise constant functions* $u(.) : \mathbb{R}_+ \to U$*, defined by the sequence* $(u_j, v_j)_{j\in\mathbb{N}}$ *of pairs of elements of* U *such that, for* $j \in \mathbb{N}$*:*

$$\forall t \in [j\theta, (j+1)\theta[, u(t) = (u_j, v_j)_{x_j,K}(t - j\theta), \tag{2.9}$$

where $x_j = x_{x_0,u(.)}(j\theta)$ *and* $(u_j, v_j)_{x_j,K}(t - j\theta)$ *refers to definition 2.3.*

DEFINITION 2.9. *The viability kernel of* K *in EDT of time step* θ *under dynamics 2.1, denoted* $Viab_{EDT}(\theta, K)$*, is the set of states* $x_0 \in K$ *for which there exists a control function* $u(.)$ *in* $\mathcal{E}_\theta(x_0, U, K)$ *such that the trajectory from* x_0 *applying control function* $u(.)$ *remains indefinitely in* K*:*

$$Viab_{EDT}(\theta, K) = \{x_0 \in K, \exists u(.) \in \mathcal{E}_\theta(x_0, U, K) \,|\, \forall t \in \mathbb{R}_+, x_{x_0,u(.)}(t) \in K\}. \tag{2.10}$$

We now define a viability domain sewed in K with time step θ under dynamics 2.1. The image of sewing is suggested by considering the trajectory $X_{x_0,(u,v)_K}(\theta)$ as a thread that should be included in K while the stitch $x_{x_0,(u,v)_K}(\theta)$ should belong to the viability domain.

DEFINITION 2.10. *A set* $E \subset K$ *is a viability domain sewed in* K *with time step* θ *under dynamics 2.1, iff:*

$$\forall x_0 \in E, \exists (u, v) \in U^2 \,|\, X_{x_0,(u,v)_K}(\theta) \subset K \text{ and } x_{x_0,(u,v)_K}(\theta) \in E. \tag{2.11}$$

PROPOSITION 2.11. *The viability kernel of K in EDT of time step θ is the largest viability domain sewed in K with time step θ.*

Proof. For all $x_0 \in Viab_{EDT}(\theta, K)$ there exists $u(.) \in \mathcal{E}_\theta(x_0, U, K)$ such that for all $t \in \mathbb{R}_+$, $X_{x_0,u(.)}(t) \subset K$. Let $u(.)$ be defined by the infinite series $\{(u_0, v_0), ..., (u_j, v_j), ...\}$ with, for $j \in \mathbb{N}$, $(u_j, v_j) \in U^2$. We have thus $X_{x_0,(u_0,v_0)_K}(\theta) \subset K$. Let $x_1 = x_{x_0,(u_0,v_0)_K}(\theta)$. The function $u'(.)$ defined by $\{(u_1, v_1), ..., (u_j, v_j), ...\}$ is such that for all $t \in \mathbb{R}_+$, $X_{x_1,u'(.)}(t) \subset K$. Therefore, $x_1 \in Viab_{EDT}(\theta, K)$. Therefore $Viab_{EDT}(\theta, K)$ is a viability domain sewed in K with time step θ.

Let $E \subset K$ be a viability domain sewed in K with time step θ. Consider $x_0 \in E$. By definition, there exist $(u_0, v_0) \in U^2$ such that $X_{x_0,(u_0,v_0)_K}(\theta) \subset K$ and $x_{x_0,(u_0,v_0)_K}(\theta) \in E$.

Consider now, for $j \in \mathbb{N}$, $\{(u_0, v_0), ..., (u_{j-1}, v_{j-1})\} \in (U^2)^j$, such that, denoting $x_{p+1} = x_{x_p,(u_p,v_p)_K}(\theta)$, for $p \in \{0, ..., j-1\}$, we have: $X_{x_p,(u_p,v_p)_K}(\theta) \subset K$ and $x_{x_p,(u_p,v_p)_K}(\theta) \in E$. In particular, $x_j = x_{x_{j-1},(u_{j-1},v_{j-1})_K}(\theta) \in E$. Therefore there exist $(u_j, v_j) \in U^2$ such that $x_{j+1} = x_{x_j,(u_j,v_j)_K}(\theta) \in E$ and $X_{x_j,(u_j,v_j)_K}(\theta) \subset K$ because E is a viability domain sewed in K.

Therefore, there exists $u(.) \in \mathcal{E}(x_0, U, K)$ such that for all $t \in \mathbb{R}_+$, $X_{x_0,u(.)}(t) \subset K$, hence $x_0 \in Viab_{EDT}(\theta, K)$. Therefore $E \subset Viab_{EDT}(\theta, K)$. □

2.2.2. Backstop viability kernel sewed in K.

DEFINITION 2.12. *$V \subset K$ is a backstop viability domain sewed in K with time step θ under dynamics 2.1 iff there exists a set $B \subset V$ such that:*

- *$D(B, K - V) > 0$;*
- *for all $x_0 \in B$, there exists $(u, v) \in U^2$ such that $X_{x_0,(u,v)_K}(\theta) \in K$ and $x_{x_0,(u,v)_K}(\theta) \in V$;*
- *For all $x_0 \in V - B$, there exist $(u, v) \in U^2$, such that:*
  - *either $\forall j \in \mathbb{N}$, $X_{x_0,(u,v)_K}(j\theta) \in K$ and $x_{x_0,(u,v)_K}(j\theta) \in V$,*
  - *or $\exists j \in \mathbb{N}^*$, $X_{x_0,(u,v)_K}(j\theta) \subset K$ and $x_{x_0,(u,v)_K}(j\theta) \in B$ and $\forall p < j$, $x_{x_0,(u,v)_K}(p\theta) \in V$.*

B is called a backstop of V in EDT time step θ and $D(B, K - V)$ is called the catch width of the backstop V.

It can immediately be seen that a backstop viability domain sewed in K with time step θ is a viability domain sewed in K with time step θ.

PROPOSITION 2.13. *For $(\delta, \theta) \in (\mathbb{R}_+^*)^2$, the union of all backstop viability domains sewed in K with time step θ including a backstop of catch width greater or equal to δ is itself a backstop viability domain sewed in K with time step θ including a backstop of catch width greater or equal to δ. It is called the backstop viability kernel sewed in K with time step θ of catch width δ and denoted $BSViab_{EDT}(\delta, \theta, K)$.*

Proof. The proof is similar to the one of proposition 2.6. □

2.3. Connection between backstop viability kernels in continuous and in extended discrete time.

PROPOSITION 2.14. *Let V_θ be a backstop viability domain sewed in K with time step θ under dynamics 2.1, holding set B as a backstop of catch width $\delta \in \mathbb{R}_+^*$. There exists a backstop viability domain V of K holding B as a backstop for time threshold θ such that:*

$$V_\theta \subset V. \tag{2.12}$$

Proof. Let B be a backstop of V_θ of catch width δ and let $\mathcal{C}(B)$ be the set of points $x_0 \in K$ for which there exists $(u,v) \in U^2$ such that:

- either for all $t \in \mathbb{R}_+$, $X_{x_0,(u,v)_K}(t) \subset K$;
- or there exists $t \in \mathbb{R}_+^*$, $t \geq \theta$ such that $X_{x_0,(u,v)_K}(t) \subset K$ and $x_{x_0,(u,v)_K}(t) \in B$;

Let $V = B \cup \mathcal{C}(B)$. For all $x_0 \in V_\theta - B$, $x_0 \in \mathcal{C}(B)$, therefore, $V_\theta \subset V$. Therefore $\inf_{x_0 \in B} D(x_0, K - V) \geq \delta$. V is thus a backstop viability domain holding B as a backstop of catch width greater or equal to δ, for time threshold θ. ∎

COROLLARY 2.15. *For any $(\delta,\theta) \in (\mathbb{R}_+^*)^2$, we have:*

$$BSViab_{EDT}(\delta,\theta,K) \subset BSViab(\delta,\theta,K).$$

Proof. The proof comes directly from proposition 2.14 and from the definitions of $BSViab(\delta,\theta,K)$ and $BSViab_{EDT}(\delta,\theta,K)$. ∎

PROPOSITION 2.16. *Let V be a backstop viability domain of K under dynamics 2.1, holding a backstop B of catch width δ for time threshold θ and let:*

$$M = \max\{\|\varphi(x,u)\|, x \in K, u \in U\}. \tag{2.13}$$

For all $\theta' \leq \min(\theta, \frac{\delta}{2M})$, V is a backstop viability domain sewed in K with time step θ' holding a backstop of catch width $\frac{\delta}{2}$.

Proof. Consider $\theta' \in \mathbb{R}_+^*$ such that $\theta' \leq \min(\theta, \frac{\delta}{2M})$. Let $B' = \{x \in V, D(x, K - V) \geq \frac{\delta}{2}\}$. For all $x \in B$, $D(x, K - B') \geq \frac{\delta}{2}$. For $x_0 \in B'$, because $\theta' \leq \theta$, there exists $(u,v) \in U^2$ such that $X_{x_0,(u,v)_K}(\theta') \subset K$ and $x_{x_0,(u,v)_K}(\theta') \in V$. Consider $x_0 \in V - B'$. Because $B \subset B'$, $x_0 \in V - B$. Therefore, because B is a backstop in V, there exists $(u,v) \in U^2$ such that:

- Either for all $t \in \mathbb{R}_+$, $x_{x_0,(u,v)_K}(t) \in V$, then $\forall j \in \mathbb{N}$, $x_{x_0,(u,v)_K}(j\theta') \in V$, and $X_{x_0,(u,v)_K}(j\theta') \subset V$, thus $X_{x_0,(u,v)_K}(j\theta') \subset K$,
- Or there exists $t > 0$, $t \geq \theta \mid X_{x_0,(u,v)_K}(t) \subset V$ and $y = x_{x_0,(u,v)_K}(t) \in B$. Then $y \in B'$ because $B \subset B'$. We know that $D(y, V - B') \geq \frac{\delta}{2}$ which implies that $t > \theta'$ and for all $t' \in [t - \theta', t]$, $x_{x_0,(u,v)_K}(t') \in B'$. Therefore, taking j as the integer part of $\frac{t}{\theta'}$ ensures $j\theta' \in [t - \theta', t]$ thus $x_{x_0,(u,v)_K}(j\theta') \in B'$.

Therefore, V is a backstop viability domain in EDT of time step θ' holding B' as a backstop of catch width $\frac{\delta}{2}$. ∎

COROLLARY 2.17. *Let M be defined by equation 2.13. For any $(\delta,\theta) \in (\mathbb{R}_+^*)^2$, for any $(\delta',\theta') \in (\mathbb{R}_+^*)^2$ such that $\delta' \leq \frac{\delta}{2}$ and $\theta' \leq \min(\theta, \frac{\delta}{2M})$, we have:*

$$BSViab(\delta,\theta,K) \subset BSViab_{EDT}(\delta',\theta',K).$$

Proof. The proof comes directly from proposition 2.16 and from the definitions of $BSViab(\delta,\theta,K)$ and $BSViab_{EDT}(\delta',\theta',K)$. ∎

PROPOSITION 2.18. *Let M be defined by equation 2.13. If $BSViab(K)$, the backstop viability kernel of K exists and $BSViab(K) = BSViab(\delta_0,\theta_0,K)$, for $(\delta_0,\theta_0) \in (\mathbb{R}_+^*)^2$, then:*

$$\forall \delta \leq \frac{\delta_0}{2}, \forall \theta \leq \min(\theta_0, \frac{\delta_0}{2M}),\ BSViab_{EDT}(\delta,\theta,K) = BSViab(K). \tag{2.14}$$

Proof. The proposition is a direct consequence of corollaries 2.15 and 2.17. ∎

Proposition 2.18 provides a connection between the backstop viability kernel and backstop viability kernels sewed in K (in EDT) which is important for the approximation algorithm described in the next section.

3. Approximating backstop viability kernels.

3.1. Algorithm and its basic properties. Algorithm 3.1 takes as input: $n \in \mathbb{N}$ the number of points by axis of the grid, $\delta \in \mathbb{R}_+^*$ a catch width value, $\theta \in \mathbb{R}_+^*$ a time step value and $j_A \in \mathbb{N}$ a maximum number of tested steps. It uses algorithm $\mathcal{A}$ approximating sets (or classification functions, this is equivalent) from a finite sample $\Gamma(n)$ of points in K. This sample is based on a regular grid of n^d points (n points by axis) covering K for the nearest vertex approximation and a more elaborated sample drawn on edges of the cubes defined by the grid for resistars (which are used in our tests of the algorithm reported in section 4). Other set approximation algorithms such as decision trees or support vector machines could be chosen. Considering a set $E \subset K$, $\mathcal{A}(\Gamma(n) \cap E)$ denotes the approximation of E derived from sample $\Gamma(n)$ by algorithm $\mathcal{A}$.

Algorithm 3.1 builds sets $H^i \cap \Gamma(n)$ from which are derived $\hat{H}^i = \mathcal{A}(H^i \cap \Gamma(n))$, the iterative approximations of the backstop viability kernel, with sets B^i_{in} as the approximation of their backstops. At each iteration, some points of the sample are eliminated from the definition of the next approximation if they do not pass `testInBstop` (for points in B^i_{in}) or `testInCatch` (for points in $H^i - B^i_{in}$). Under some assumptions (see further), the sets B^i_{out} contain the sets $\hat{H}^i$, thus the sets $B^i_{out} - B^i_{in}$ are wide approximations of the catches. Actually, a point passes `testInCatch` if there exists a constant control with one possible bounce that defines a trajectory sewed in K with stitches in B^i_{out} during j_A time steps, or reaching B_{in} in less than j_A time steps, without getting out from B^i_{out}. As will be shown further, in some conditions, at the last iteration of the algorithm, this test determines with full accuracy if a point belongs to the backstop viability kernel, independently from the precision of the approximation $\hat{H}^i$. The last iteration is reached when no point of the sample is eliminated. The algorithm also stops if $H^1 \cap \Gamma(n) = K \cap \Gamma(n)$.

The algorithm uses the following definitions:

- For any set $H \subset K$ and any $\delta \in \mathbb{R}_+$, the sets $(H)_\delta$ and $(H)_{-\delta}$ are respectively the erosion and the dilatation of size δ in K:

$$(H)_\delta := \{x \in H \,|\, D(x, K - H) \geq \delta\},$$
$$(H)_{-\delta} := \{x \in K \,|\, D(x, H) \leq \delta\}.$$

- The sets B^i_{in} and B^i_{out} approximate respectively $(\hat{H}^i)_\delta$ and $(\hat{H}^i)_{-\delta}$:

$$B^i_{in} := \mathcal{A}((\hat{H}^i)_\delta \cap \Gamma(n)),$$
$$B^i_{out} := \mathcal{A}((\hat{H}^i)_{-\delta} \cap \Gamma(n)).$$

- The sets H^i are recursively defined by:

$$H^1 \leftarrow \{x_0 \in K \,|\, \exists (u, v) \in U^2, X_{x_0,(u,v)_K}(\theta) \subset K\},$$
$$H^{i+1} \leftarrow \{x_0 \in B^i_{in} \,|\, \texttt{testInBstop}(x_0, B^i_{in}, B^i_{out})\} \cup$$
$$\{x_0 \in H^i \,|\, x_0 \notin B^i_{in} \text{ and } \texttt{testInCatch}(x_0, B^i_{in}, B^i_{out})\}.$$

Algorithm 3.1 `BSVApp`(n, δ, θ, j_A). Backstop viability kernel approximation.

Input: $n \in \mathbb{N}$ number of points by grid axis, $\delta \in \mathbb{R}_+^*$ width of backstop catch, $\theta \in \mathbb{R}_+^*$ time step, $j_A \in \mathbb{N}$ number of tested steps.

$H^1 \cap \Gamma(n) \leftarrow \{\xi \in \Gamma(n), \exists (u,v) \in U^2, X_{\xi,(u,v)_K}(\theta) \subset K\}$;
if $H^1 \cap \Gamma(n) = K \cap \Gamma(n)$ **then**
  return K;
end
$\hat{H}^1 \leftarrow \mathcal{A}(H^1 \cap \Gamma(n))$;
$i \leftarrow 0$;
repeat
  $i \leftarrow i + 1$;
  $B_{in}^i \leftarrow \mathcal{A}((\hat{H}^i)_\delta \cap \Gamma(n))$; $B_{out}^i \leftarrow \mathcal{A}((\hat{H}^i)_{-\delta} \cap \Gamma(n))$;
  $H^{i+1} \cap \Gamma(n) \leftarrow \{\xi \in B_{in}^i \cap \Gamma(n) \,|\, \texttt{testInBstop}(\xi, B_{in}^i, B_{out}^i)\} \cup$
    $\{\xi \in H^i \cap \Gamma(n) \,|\, \xi \notin B_{in}^i \text{ and } \texttt{testInCatch}(\xi, B_{in}^i, B_{out}^i)\}$
  $\hat{H}^{i+1} \leftarrow \mathcal{A}(H^{i+1} \cap \Gamma(n))$;
until $H^i \cap \Gamma(n) = H^{i+1} \cap \Gamma(n)$;
return $(\hat{H}^i)$;

Algorithm 3.2 `testInCatch`$(x_0, B_{in}^i, B_{out}^i)$. Test selecting points in $B_{out}^i - B_{in}^i$.

Input: $x_0 \in B_{out}^i - B_{in}^i$, B_{in}^i , B_{out}^i.

$$b \leftarrow \exists (u,v) \in U^2 \,|\, \text{either } \forall j \le j_A, X_{x_0,(u,v)_K}(j\theta) \subset K \text{ and } x_{x_0,(u,v)_K}(j\theta) \in B_{out}^i$$
$$\text{or } \exists j < j_A \begin{cases} X_{x_0,(u,v)_K}(j\theta) \subset K \text{ and } x_{x_0,(u,v)_K}(j\theta) \in B_{in}^i \\ \text{and } \forall p < j, x_{x_0,(u,v)_K}(p\theta) \in B_{out}^i. \end{cases}$$

return (b);

Algorithm 3.3 `testInBstop`$(x_0, B_{in}^i, B_{out}^i)$. Test selecting points in B_{in}^i.

Input: $x_0 \in B_{in}^i$, B_{in}^i, B_{out}^i.

$$b \leftarrow \exists (u,v) \in U^2 \,|\, X_{x_0,(u,v)_K}(\theta) \subset K \text{ and } y \leftarrow x_{x_0,(u,v)_K}(\theta) \,|$$
$$\begin{cases} y \in B_{in}^i \text{ or} \\ y \in B_{out}^i - B_{in}^i \text{ and } \texttt{testInCatch}(y, B_{in}^i, B_{out}^i) \end{cases}$$

return (b);

The basic properties of algorithm 3.1 require the following assumption.

ASSUMPTION 1. *For any $\epsilon \in \mathbb{R}_+^*$, there exists $n_1(\epsilon) \in \mathbb{N}^*$, such that for all $n > n_1(\epsilon)$, at any iteration i of the algorithm, for any set $H \in \{H^i, (\hat{H}^i)_\delta, (\hat{H}^i)_{-\delta}\}$, we have:*

$$(H)_\epsilon \subset \hat{H} \subset (H)_{-\epsilon} \text{ and } (\hat{H})_\epsilon \subset H \subset (\hat{H})_{-\epsilon},$$

where $\hat{H} := \mathcal{A}(H \cap \Gamma(n))$.

PROPOSITION 3.1. *Under assumption 1, for all $n > n_1(\frac{\delta}{4})$, at each iteration of algorithm 3.1, we have:*

$$(H^i)_{\frac{3\delta}{2}} \subset B_{in}^i \subset (H^i)_{\frac{\delta}{2}} \text{ and } (H^i)_{-\frac{\delta}{2}} \subset B_{out}^i \subset (H^i)_{\frac{-3\delta}{2}}.$$

Proof. For $\epsilon \in \mathbb{R}_+^*$, $\epsilon < \frac{\delta}{2}$, for $n > n_1(\epsilon)$, at any iteration i, we have:

$$(H^i)_\epsilon \subset \hat{H}^i$$

This implies:

$$((H^i)_\epsilon)_\delta \subset (\hat{H}^i)_\delta \text{ hence: } (H^i)_{\delta+\epsilon} \subset (\hat{H}^i)_\delta. \tag{3.1}$$

and assumption 1 ensures also:

$$(\hat{H}^i)_\epsilon \subset H^i,$$

which implies:

$$((\hat{H}^i)_\epsilon)_{\delta-\epsilon} \subset (H^i)_{\delta-\epsilon} \text{ hence: } (\hat{H}^i)_\delta \subset (H^i)_{\delta-\epsilon}. \tag{3.2}$$

Therefore:

$$(H^i)_{\delta+\epsilon} \subset (\hat{H}^i)_\delta \subset (H^i)_{\delta-\epsilon}. \tag{3.3}$$

Applying assumption 1 to set $(\hat{H}^i)_\delta$, and setting $B_{in} = \mathcal{A}((\hat{H}^i)_\delta \cap \Gamma(n))$ we get:

$$\begin{aligned}((\hat{H}^i)_\delta)_\epsilon \subset B^i_{in} \subset ((\hat{H}^i)_\delta)_{-\epsilon} \text{ hence (eq: 3.3):}\\ ((H^i)_{\delta+\epsilon})_\epsilon \subset B^i_{in} \subset ((H^i)_{\delta-\epsilon})_{-\epsilon} \text{ hence:}\\ (H^i)_{\delta+2\epsilon} \subset B^i_{in} \subset (H^i)_{\delta-2\epsilon}.\end{aligned}$$

Applying the same reasoning to B^i_{out} (*mutatis mutandis*) and taking $\epsilon = \frac{\delta}{4}$ completes the proof. ∎

PROPOSITION 3.2. *Under assumption 1, for $n > n_1(\frac{\delta}{4})$, algorithm 3.1 stops after a finite number of iterations.*

Proof. Proposition 3.1 guarantees that, for $n > n_1(\frac{\delta}{4})$, at each iteration i of algorithm 3.1, we have: $B^i_{in} \subset H^i$ and thus $H^{i+1} \cap \Gamma(n) \subset H^i \cap \Gamma(n)$. Since $\Gamma(n)$ is finite, the procedure reaches $H^{i+1} \cap \Gamma(n) = H^i \cap \Gamma(n)$ after a finite number of iterations. ∎

PROPOSITION 3.3. *Under assumption 1, for $n > n_1(\frac{\delta}{4})$, at any step i of algorithm 3.1 we have:*

$$BSViab_{EDT}(\frac{3\delta}{2}, \theta, K) \subset H^i. \tag{3.4}$$

Proof. Let $V = BSViab_{EDT}(\frac{3\delta}{2}, \theta, K)$. Obviously $V \subset H^1$.
Assume now $V \subset H^i$ and consider $x_0 \in V$. Two cases occur:

- $x_0 \in V - (V)_{\frac{3\delta}{2}}$, then there exists $(u, v) \in U^2$ such that:
  - either for all $j \in \mathbb{N}$, $X_{x_0,(u,v)_K}(j\theta) \subset K$ and $x_{x_0,(u,v)_K}(j\theta) \in V$, then for all $j \in \mathbb{N}$, $x_{x_0,(u,v)_K}(j\theta) \in B^i_{out}$, because $V \subset H^i$ and because of proposition 3.1, $H^i \subset B^i_{out}$;
  - or there exists $j \in \mathbb{N}^* \mid X_{x_0,(u,v)_K}(j\theta) \subset K$ and $x_{x_0,(u,v)_K}(j\theta) \in (V)_{\frac{3\delta}{2}}$ and for all $p < j$, $x_{x_0,(u,v)_K}(p\theta) \in V$, then $x_{x_0,(u,v)_K}(j\theta) \in B^i_{in}$, (because $V \subset H^i$, $(V)_{\frac{3\delta}{2}} \subset (H^i)_{\frac{3\delta}{2}}$ and because of proposition 3.1, $H^i_{\frac{3\delta}{2}} \subset B^i_{in}$) and for all $p < j$, $x_{x_0,(u,v)_K}(p\theta) \in B^i_{out}$.

Therefore $\texttt{testInCatch}(x_0, B^i_{in}, B^i_{out}) = \texttt{true}$. If $x_0 \notin B^i_{in}$, then $x_0 \in H^{i+1}$. If $x_0 \in B^i_{in}$, this implies $\texttt{testInBstop}(x_0, B^i_{in}, B^i_{out}) = \texttt{true}$, thus $x_0 \in H^{i+1}$.

- $x_0 \in (V)_{\frac{3\delta}{2}}$, then $x_0 \in B^i_{in}$ (because of 3.1) and there exists $(u, v) \in U^2$ such that $X_{x_0,(u,v)_K}(\theta) \subset K$ and $y = x_{x_0,(u,v)_K}(\theta) \in V$. Therefore $y \in B^i_{out}$. If $y \in V - (V)_{\frac{3\delta}{2}}$, and as shown in the previous case, $\texttt{testInCatch}(y, B^i_{in}, B^i_{out}) = \texttt{true}$. Else $y \in (V)_{\frac{3\delta}{2}}$ hence $y \in B^i_{in}$ therefore $\texttt{testInBstop}(x_0, B^i_{in}, B^i_{out}) = \texttt{true}$.

Finally, in all cases, $x_0 \in H^{i+1}$, therefore $V \subset H^{i+1}$. ∎

3.2. Inclusion of the algorithm output in viability kernel in EDT. The next propositions require assumption 2 and a new definition.

ASSUMPTION 2. *For any $\epsilon \in \mathbb{R}^*_+$, there exists $n_2(\epsilon) \in \mathbb{N}$, such that for all $n > n_2(\epsilon)$, at any iteration i of the algorithm, for any set H^i, we have:*

$$\forall x \in K - H^i, \exists \xi \in (K - H^i) \cap \Gamma(n), D(x, \xi) < \epsilon,$$

DEFINITION 3.4. *For $j \in \mathbb{N}$, let $\mathcal{H}^j$ be the set of points of K for which the exit time when applying the EDT control of time step θ is greater than $j\theta$:*

$$\mathcal{H}^j := \{x_0 \in K, \exists u(.) \in \mathcal{E}_\theta(x_0, U, K), \tau^\sharp_K(x_0, u(.)) \geq j\theta\}. \tag{3.5}$$

We start by showing that the final set H^q defined by the algorithm is included in $\mathcal{H}^{j_A}$. Then we distinguish the case where there exists $j_A \in \mathbb{N}$ such that $\mathcal{H}^{j_A} = Viab_{EDT}(\theta, K)$ from the case where there is none.

PROPOSITION 3.5. *Under assumptions 1 and 2, for $n > \max(n_1(\frac{\delta}{4}), n_2(\frac{\delta}{2}))$, at the last iteration q of algorithm 3.1, we have:*

$$H^q \subset \mathcal{H}^{j_A}.$$

Proof. At iteration i of algorithm 3.1, let $\mathcal{F}^i = B^i_{in} \cap (K - \mathcal{H}^{j_A})$. Assume $\mathcal{F}^i \neq \emptyset$ and, for all $x_0 \in \mathcal{F}^i$, $\texttt{testInBstop}(x_0, B^i_{in}, B^i_{out}) = \texttt{true}$. For each point $x_0 \in \mathcal{F}^i$, we define $U(x_0) \subset U^4$ and $j(x_0) \in \mathbb{N}$ as follows:

- If there exists $(u, v) \in U^2$ such that $X_{x_0,(u,v)_K}(\theta) \subset K$ and $x_{x_0,(u,v)_K}(\theta) \in B^i_{in}$. Then $U(x_0) := \emptyset$ and $j(x_0) := 0$;
- Else there exists $(u, v) \in K^2 \mid X_{x_0,(u,v)_K}(\theta) \subset K$ and $y := x_{x_0,(u,v)_K}(\theta) \in B^i_{out} - B^i_{in}$, and, because $x_0 \notin \mathcal{H}^{j_A}$, there exists $(u', v') \in U^2$ and $j \in \mathbb{N}$, $j < j_A$ such that $X_{y,(u',v')_K}(j\theta) \subset K$, $x_{y,(u',v')_K}(j\theta) \in B^i_{in}$ and for $p \in \mathbb{N}$, $0 < p < j$, $x_{y,(u',v')_K}(p\theta) \in B^i_{out}$. Then $U(x_0) := (u, v, u', v')$ and $j(x_0) := j$.

We define set E as follows:

$$E := \mathcal{F}^i \cup \{x_{x_{x_0,(u,v)_K}(\theta),(u',v')_K}(p\theta), x_0 \in \mathcal{F}^i, (u, v, u', v') = U(x_0) \neq \emptyset, 0 \leq p \leq j(x_0)\}.$$

It can easily be verified that E is a viability domain sewed in K with time step θ. Therefore $E \subset \mathcal{H}^{j_A}$. This is impossible because $\mathcal{F}^i \subset E$ and $\mathcal{F}^i \subset (K - \mathcal{H}^{j_A})$, by definition.

Therefore, there exists $x_0 \in \mathcal{F}^i$, such that $\texttt{testInBstop}(x_0, B^i_{in}, B^i_{out}) = \texttt{false}$, thus $x_0 \in K - H^{i+1}$. Because of assumption 2, there exists $\xi \in (K - H^{i+1}) \cap \Gamma(n)$ such that $D(x_0, \xi) < \frac{\delta}{2}$ therefore, $\xi \in H^i$ because $x_0 \in B^i_{in}$ and assumption 1 ensures $B^i_{in} \subset (H^i)_{\frac{\delta}{2}}$, hence $D(x_0, K - H^i) \geq \frac{\delta}{2}$.

Consider now the case when $\mathcal{F}^i = \emptyset$, hence $B^i_{in} \subset \mathcal{H}^{j_A}$, and consider a point $x_0 \in H^i$ such that $x_0 \notin \mathcal{H}^{j_A}$. For $(u,v) \in U^2$, let $y = x_{x_0,(u,v)_K}(\theta)$. $y \notin \mathcal{H}^{j_A}$ (otherwise we would have $x_0 \in \mathcal{H}^{j_A}$), therefore $y \notin B^i_{in}$. Assume $y \in B^i_{out} - B^i_{in}$. Then, for any $(u',v') \in U^2$, $X_{y,(u',v')_K}(j_A\theta) \subset K$ is impossible because $x_0 \notin \mathcal{H}^{j_A}$. Suppose that there exist $(u',v') \in U^2$ and $j \leq j_A$, such that $X_{y,(u',v')_K}(j\theta) \subset K$ and $x_{y,(u',v')_K}(j\theta) \in B^i_{in}$. $B^i_{in} \subset \mathcal{H}^{j_A}$ implies $x_{y,(u',v')_K}(j\theta) \in \mathcal{H}^{j_A}$ which is impossible because $x_0 \notin \mathcal{H}^{j_A}$. Therefore, `testInBstop`$(x_0, B^i_{in}, B^i_{out})$= `false` and, for all $x_0 \in H^i$ such that $x_0 \notin \mathcal{H}^{j_A}$, $x_0 \notin H^{i+1}$. Therefore $H^{i+1} \subset \mathcal{H}^{j_A}$.

To summarize, while B^i_{in} is not included in $\mathcal{H}^{j_A}$, $H^{i+1} \cap \Gamma(n) \neq H^i \cap \Gamma(n)$ hence the algorithm does not stop and at the first iteration i such that $B^i_{in} \subset \mathcal{H}^{j_A}$, we have $H^{i+1} \subset \mathcal{H}^{j_A}$. Assumption 1 ensures that the algorithm stops after a finite number of iterations q and that $H^q \subset H^{i+1} \subset \mathcal{H}^{j_A}$. ∎

DEFINITION 3.6. *The supremum of steps that the system's trajectory remains in K when applying the EDT control of time step θ from points of K which are not viable for the EDT control of time step θ, is:*

$$j^\sharp_{EDT}(\theta,K) := \sup_{x_0 \in K - Viab_{EDT}(\theta,K)} \{j \in \mathbb{N}, \exists u(.) \in \mathcal{E}_\theta(x_0,U,K), X_{x_0,u(.)}(j\theta) \subset K\}.$$

For $n \in \mathbb{N}$, the same supremum restricted to $(K - Viab_{EDT}(\theta,K)) \cap \Gamma(n)$ is:

$$j^\sharp_{EDT}(n,\theta,K) := \sup_{\xi \in (K - Viab_{EDT}(\theta,K)) \cap \Gamma(n)} \{j \in \mathbb{N}, \exists u(.) \in \mathcal{E}_\theta(\xi,U,K), X_{\xi,u(.)}(j\theta) \subset K\}.$$

PROPOSITION 3.7. *Under assumptions 1 and 2, for $n > \max(n_1(\frac{\delta}{4}), n_2(\frac{\delta}{2}))$, and if $j^\sharp_{EDT}(\theta,K) < \infty$ and $j_A > j^\sharp_{EDT}(\theta,K)$, at the last step q of algorithm 3.1 we have:*

$$H^q \subset Viab_{EDT}(\theta,K). \tag{3.6}$$

Proof. The proof comes directly from proposition 3.5 and from the fact that $\mathcal{H}^{j_A} = Viab_{EDT}(\theta,K)$ because $j_A > j^\sharp_{EDT}(\theta,K)$. ∎

PROPOSITION 3.8. *Assume $j^\sharp_{EDT}(n,\theta,K) < \infty$. Under assumptions 1 and 2, for $n > \max(n_1(\frac{\delta}{4}), n_2(\frac{\delta}{2}))$, and if $j_A > j^\sharp_{EDT}(n,\theta,K)$, the last iteration $q \in \mathbb{N}$ of algorithm 3.1 defines set H^q such that:*

$$(H^q \cap \Gamma(n)) \subset (Viab_{EDT}(\theta,K) \cap \Gamma(n)). \tag{3.7}$$

Proof. Proposition 3.5 ensures that for all $\xi \in \Gamma(n)$, if $\xi \in H^q$ then $\xi \in \mathcal{H}^{j_A}$. Because $j_A > j^\sharp_{EDT}(n,\theta,K)$, $\xi \in \mathcal{H}^{j_A}$ implies $\xi \in Viab_{EDT}(\theta,K)$. ∎

3.3. Convergence to the backstop viability kernel. The following proposition requires a new assumption and a new definition.

ASSUMPTION 3. *There exists $(\theta_0,\delta_0) \in (\mathbb{R}^*_+)^2$ such that $BSViab(\delta_0,\theta_0,K) = BSViab(K)$ is the backstop viability kernel of K.*

DEFINITION 3.9. *Under assumption 3, the maximum number of time steps in K for constant control with one possible bounce starting from points in $K - BSViab(K)$, denoted $j^\sharp_C(\theta,K)$ is:*

$$j^\sharp_C(\theta,K) = \sup_{x_0 \in K - BSViab(K)} \{j \in \mathbb{N}, \exists (u,v) \in U^2, X_{x_0,(u,v)_K}(j\theta) \subset K\}. \tag{3.8}$$

PROPOSITION 3.10. *Let M be defined by equation 2.13. If:*

- *Assumptions 1, 2 and 3 are satisfied and $n > \max(n_1(\frac{\delta}{4}), n_2(\frac{\delta}{4}))$,*
- *$j_C^\sharp(\theta, K) < \infty$ and $j_A > j_C^\sharp(\theta, K)$,*

then for all $\delta \in \mathbb{R}_+^*$ such that $\frac{3\delta}{2} \leq \frac{\delta_0}{2}$, for all $\theta \leq \min(\theta_0, \frac{\delta_0}{2M})$, at the last step q of algorithm 3.1 we have:

$$H^q = BSViab(K).$$

Proof. The structure of the proof is similar to the one of proposition 3.5. Let $V = BSViab(K)$. Because of proposition 2.18, the choice of δ and θ ensures $V = BSViab_{EDT}(\frac{\delta}{4}, \theta, K)$. Suppose $(H^i)_{\frac{\delta}{4}} \cap (K - V) \neq \emptyset$ and that:

- for all points $x_0 \in (H^i)_{\frac{\delta}{4}} - B_{in}^i$, $\texttt{testInCatch}(x_0, B_{in}^i, B_{out}^i) = \texttt{true}$ and
- for all points $x_0 \in B_{in}^i$, $\texttt{testInBstop}(x_0, B_{in}^i, B_{out}^i) = \texttt{true}$.

For $x_0 \in (H^i)_{\frac{\delta}{4}}$ we define $V(x_0) \in U^2$ or $U(x_0) \in U^4$ and $j(x_0) \in \mathbb{N}$ as follows:

- If $x_0 \in (H^i)_{\frac{\delta}{4}} - B_{in}^i$, there exists $(u, v) \in U^2$, and $j \in \mathbb{N}$, $j \leq j_C^\sharp(\theta, K) < j_A$, such that $X_{x_0,(u,v)_K}(j\theta) \subset K$, $x_{x_0,(u,v)_K}(j\theta) \in B_{in}^i$ and for all $p < j$, $x_{x_0,(u,v)_K}(p\theta) \in B_{out}^i$. Then, $V(x_0) := (u, v)$ and $j(x_0) := j$,
- If $x_0 \in B_{in}^i$, because $j_C^\sharp(\theta, K) < j_A$, we can define $U(x_0)$ and $j(x_0)$ like we did in the proof of proposition 3.5.

Let E be defined as follows:

$$E := (H^i)_{\frac{\delta}{4}} \cup$$
$$\{x_{x_0,(u,v)_K}(p\theta), x_0 \in (H^i)_{\frac{\delta}{4}} - B_{in}^i, (u, v) = V(x_0), 0 < p \leq j(x_0)\} \cup$$
$$\{x_{x_{x_0,(u,v)_K}(\theta),(u',v')_K}(p\theta), x_0 \in B_{in}^i, (u, v, u', v') = U(x_0) \neq \emptyset, 0 \leq p \leq j(x_0)\}.$$

By construction, E is a backstop viability domain sewed in K with time step θ holding B_{in}^i as a backstop of catch width greater or equal to $\frac{\delta}{4}$. E is not included in V because $E \subset (H^i)_{\frac{\delta}{4}}$ and by hypothesis, $(H^i)_{\frac{\delta}{4}} \cap (K - V) \neq \emptyset$. This is impossible because the choice of θ ensures that $V = BSViab_{EDT}(\frac{\delta}{4}, \theta, K)$ and thus includes all backstop viability domains sewed in K of catch width greater or equal to $\frac{\delta}{4}$ with time step θ, by definition.

Therefore, there exists $x_0 \in (H^i)_{\frac{\delta}{4}} - B_{in}^i$, such that $\texttt{testInCatch}(x_0, B_{in}^i, B_{out}^i) = \texttt{false}$, or there exists $x_0 \in B_{in}^i$, such that $\texttt{testInBstop}(x_0, B_{in}^i, B_{out}^i) = \texttt{false}$. In both cases, there exists a point $x_0 \in (H^i)_{\frac{\delta}{4}}$ thus such that $D(x_0, K - H^i) \geq \frac{\delta}{4}$ and $x_0 \notin H^{i+1}$. Because of assumption 2, there exists $\xi \in \Gamma(n)$ such that $\xi \notin H^{i+1}$ and $D(x_0, \xi) < \frac{\delta}{4}$, therefore $\xi \in H^i$. Therefore $H^i \cap \Gamma(n) \neq H^{i+1} \cap \Gamma(n)$.

Suppose now $(H^i)_{\frac{\delta}{4}} \subset V$. Consider $x_0 \in H^i - V$. $x_0 \in H^i - B_{in}^i$ because $B_{in}^i \subset (H^i)_{\frac{\delta}{4}}$. Suppose $\texttt{testInCatch}(x_0, B_{in}^i, B_{out}^i) = \texttt{true}$. Then there exist $(u, v) \in U^2$ and $j \in \mathbb{N}$, $j \leq j_C^\sharp(\theta, K) < j_A \mid X_{x_0,(u,v)_K}(j\theta) \subset K$ and $x_{x_0,(u,v)_K}(j\theta) \in B_{in}^i$ and for all $p < j$, $x_{x_0,(u,v)_K}(p\theta) \in B_{out}^i$. Then the set $V \cup \{x_{x_0,(u,v)_K}(p\theta), p \in \{0, .., j\}\}$ is a backstop viability domain of time threshold θ admitting B_{in}^i as a backstop of catch width at least $\frac{\delta}{4}$ and this backstop viability domain is not included in V. This is impossible. Therefore, $\texttt{testInCatch}(x_0, B_{in}^i, B_{out}^i) = \texttt{false}$. This implies: $H^{i+1} \subset V$. Because of proposition 2.18, the choice of δ and θ ensures $V = BSViab_{EDT}(\frac{3\delta}{2}, \theta, K)$. Proposition 3.3 can be applied because assumption 1 is satisfied thus $V \subset H^{i+1}$ and $V \subset H^{i+2}$ therefore $H^{i+1} = H^{i+2} = V$.

Overall, while $(H^i)_{\frac{\delta}{4}}$ is not included in V, $H^i \cap \Gamma(n) \neq H^{i+1} \cap \Gamma(n)$ therefore the algorithm does not stop. Once $(H^i)_{\frac{\delta}{4}} \subset V$, the algorithm stops after at most one iteration and $H^{i+1} = BSViab(K)$. ∎

3.4. Convergence rate. We now derive the implications of the previous propositions on the convergence rate of the algorithm output to the backstop viability kernel when n, the number of points by axis of the grid, increases. This requires a new assumption which uses the Hausdorff distance $D_h(A, B)$ between two sets A and B:

$$D_h(A, B) = \max(D(A, B), D(B, A)). \tag{3.9}$$

ASSUMPTION 4. *There exists a function $\epsilon(n) : \mathbb{N} \to \mathbb{R}_+$, $\epsilon(n) \to 0$ when $n \to \infty$, such that, for any set H satisfying some smoothness properties (see examples in section 4), there exists $n_4 \in \mathbb{N}$ such that for $n > n_4$:*

$$D_h(\mathcal{A}(H \cap \Gamma(n)), H) < \epsilon(n). \tag{3.10}$$

PROPOSITION 3.11. *Assume the conditions of proposition 3.10. Under assumption 4 and if $BSViab(K)$ satisfies the required smoothness conditions and n is large enough, at the final iteration q of algorithm 3.1, we have:*

$$D_h(\hat{H}^q, BSViab(K)) < \epsilon(n). \tag{3.11}$$

Proof. Let $V = BSViab(K)$. Proposition 3.10 ensures $H^q = V$ and because of assumption 4, $D_h(\mathcal{A}(V \cap \Gamma(n), V) < \epsilon(n)$. ∎

PROPOSITION 3.12. *Assume that $BSViab(K) = BSViab(\delta_0, \theta_0)$ is the backstop viability kernel and is equal to the viability kernel: $BSViab(K) = Viab(K)$. Assume that the conditions of proposition 3.8 are satisfied and $j_A > j^\sharp_{EDT}(n, \theta, K)$. Under assumption 4 and if $Viab(K)$ satisfies the required smoothness conditions and n is large enough, then for all $\delta \in \mathbb{R}^*_+$ such that $\frac{3\delta}{2} \leq \frac{\delta_0}{2}$, for all $\theta \leq \min(\theta_0, \frac{\delta_0}{2M})$, at the final iteration q of algorithm 3.1, we have:*

$$D_h(\hat{H}^q, Viab(K)) < \epsilon(n). \tag{3.12}$$

Proof. Let $V = Viab(K)$. In the considered conditions, $V = Viab_{EDT}(\theta, K)$ because $BSViab(K) \subset Viab_{EDT}(\theta, K)$ and $Viab_{EDT}(\theta, K) \subset V$. Proposition 3.8 ensures $H^q \cap \Gamma(n) = V \cap \Gamma(n)$ and because of assumption 4, $D_h(\mathcal{A}(V \cap \Gamma(n), V) < \epsilon(n)$. ∎

Note that the case $Viab(K) = BSViab(K)$ is particularly interesting, because the convergence rate $\epsilon(n)$ can be ensured even for $j^\sharp_{EDT}(\theta, K) = \infty$ and $j^\sharp_C(\theta, K) = \infty$, whereas if $Viab(K) \neq BSViab(K)$, it requires $j^\sharp_C(\theta, K) < \infty$. The next section reports tests performed in this case.

4. Tests of the convergence rate when viability kernel and backstop viability kernel are equal. In the tests, the trajectories $X_{x_0,(u,v)_K}(j\theta)$ are approximated by the Runge and Kutta method with a time step smaller than θ ensuring that several trajectory points are computed in each grid cube. Hence the approximation error on the trajectory (of the order of n^{-4}) is negligible with respect to the set approximation error (at best of the order of n^{-2}).

4.1. Tested viability problems. The tests are performed on three viability problems in d dimensions.

4.1.1. Population problem. In the original 2D version [3], $x_1(t)$ represents the size of a population, which grows or diminishes with the evolution rate $x_2(t)$ that can be modified by the control. The populations should be kept within some bounds. The system is written as follows, $m_1, M_1, m_2, M_2, m_u, M_u$ being positive parameters:

$$\begin{cases} x_1'(t) = x_1(t)x_2(t) \\ x_2'(t) = u(t), \end{cases} \quad \text{with} \quad \begin{cases} -m_u \leq u(t) \leq +M_u, \\ (x_1(t), x_2(t)) \in K = [m_1, M_1] \times [-m_2, M_2]. \end{cases} \tag{4.1}$$

In the extension to d dimensions, the model includes $d-1 \geq 2$ populations $x_1, x_2, .., x_{d-1}$ and it uses variable $\tilde{x}(t)$:

$$\tilde{x}(t) = \sum_{i=2}^{d-1} \frac{(M_1 - m_1)^2}{4} - \left(x_i(t) - \frac{M_1 + m_1}{2} \right)^2. \tag{4.2}$$

The extended viability problem is:

$$\begin{cases} x_1'(t) = y(t)\left(x_1(t) + \alpha\tilde{x}(t)\right) \\ x_i'(t) = 0, i \in \{2, .., d-1\} \\ x_d'(t) = u(t), \end{cases} \quad \text{with} \quad \begin{cases} -m_u \leq u(t) \leq M_u \\ m_1 \leq x_i(t) \leq M_1, i \in \{1, .., d-1\}, \\ -m_2 \leq x_d(t) \leq M_2. \end{cases} \tag{4.3}$$

Where α is a parameter. In this system, the dynamics of population x_1 non-linearly depends on the abundance of the other populations which are constant.

The analytical definition of the viability kernel of the d-dimensional problem can be directly derived from the one of the 2D case (provided in [3]):

$$Viab(K) = \{x \in K, \partial V_-(x) \leq x_d \leq \partial V_+(x)\}, \text{ with:} \tag{4.4}$$

$$\begin{cases} \partial V_+(x) = \sqrt{2m_u \log\left(\frac{M_x + \alpha\bar{x}}{x_1 + \alpha\bar{x}}\right)}, \\ \partial V_-(x) = -\sqrt{2M_u \log\left(\frac{x_1 + \alpha\bar{x}}{m_x + \alpha\bar{x}}\right)}. \end{cases} \tag{4.5}$$

The backstop viability kernel of this problem exists and is equal to the viability kernel.

4.1.2. Consumption problem. In the original 2D problem [3], $x_1(t)$ represents the consumption of a primary good and $x_2(t)$ a critical level of consumption above which the prices can decrease and accelerate consumption and below which, on the contrary the prices increase and decrease the consumption. The critical level x_2 can be modified by a control within some bounds in order to maintain the consumption within some bounds. The system is written as follows (m_u, M_u, m_1, M_1 and M_2 are positive):

$$\begin{cases} x_1'(t) = x_1(t) - x_2(t) \\ x_2'(t) = u(t), \end{cases} \quad \text{with} \quad \begin{cases} -m_u \leq u(t) \leq M_u, \\ (x_1(t), x_2(t)) \in K = [-m_1, M_1] \times [0, M_2]. \end{cases} \tag{4.6}$$

Like the population model, we extend the consumption model to d dimensions by considering $d-1 \geq 2$ consumption variables $x_1, x_2, .., x_{d-1}$. We also define variable $\tilde{x}(t)$ with equation 4.2. The extended system is:

(4.7)
$$\begin{cases} x_1'(t) = (x_1(t) + \alpha\tilde{x}(t)) - x_d(t) \\ x_i'(t) = 0, i = 2, .., d-1 \\ x_d'(t) = u(t), \end{cases} \quad \text{with} \quad \begin{cases} -M_u \leq u(t) \leq M_u, \\ -m_1 \leq x_i(t) \leq M_1, i \in \{1, .., d-1\}, \\ 0 \leq x_d(t) \leq M_2. \end{cases}$$

The analytical definition of the viability kernel can easily be derived from its expression in the 2D problem (provided in [3]):

$$Viab(K) = \{x \in K, \partial V_-(x) \leq x_1 \leq \partial V_+(x)\} \text{ with:} \tag{4.8}$$

$$\begin{cases} \partial V_-(x) = x_d - m_u + m_u \exp\left(\frac{-x_d}{m_u}\right) - \alpha\tilde{x}, \\ \partial V_+(x) = x_d + M_u - M_u \exp\left(\frac{x_d - M_1}{M_u}\right) - \alpha\tilde{x}. \end{cases} \tag{4.9}$$

For this problem also, it can be verified that the backstop viability kernel exists and equals the viability kernel.

4.1.3. Variants of the models with oblique trajectories. In both population and consumption problems, the fact that $x_i'(t) = 0$ for $i \in \{2, .., d-1\}$ can be seen as an easy particular case for the resistar approximation because in each 2-dimensional grid defined by axes x_1 and x_d, the problem to solve is the same as in 2 dimensions. Increasing the dimensionality requires only the interpolation by the resistars between these 2-dimensional classifications.

In order to test how the approach performs on a more difficult problem, instead of keeping all trajectories in the 2 D spaces parallel to (b_1, b_d), (b_i, $i \in \{1, .., d\}$ being the canonical basis vectors), for point $x = (x_1, ..., x_d)$ the trajectory is set in the 2D space generated by vectors $(z(x), b_d)$, with:

$$z(x) = b_1 + \sum_{i=2}^{d-1} z_i(x) b_i \text{ with, setting } m(x_1) = \frac{1-\beta}{2} + \beta x_1, \tag{4.10}$$

$$\text{for } i \in \{2, ..., d-1\}, z_i(x) = \begin{cases} \beta \frac{x_i}{m(x_1)}, \text{ if } x_i \leq m(x_1), \\ \beta \frac{1-x_i}{1-m(x_1)}, \text{ otherwise,} \end{cases} \tag{4.11}$$

where β is a parameter ($0 \leq \beta < 1$) and the problem is rescaled so that $K = [0,1]^d$. When x is located on the right line of direction $b_1 + \beta \sum_{i=2}^{d-1} b_i$ which includes point $(\frac{1-\beta}{2}, ..., \frac{1-\beta}{2})$, it makes the maximum angle with b_1. The components $z_i(x)$ equal 0 for $x_i = 0$ or $x_i = 1$.

This leads to population or consumption problems in the spaces $(z(x), b_d)$ with a constraint set equal to $[0, \|z(x)\|] \times [0,1]$ instead of $[0,1] \times [0,1]$. Their viability kernel in the plane $(z(x), b_d)$ can be derived directly from the viability kernels of the 2D problems.

4.1.4. Spirals problem. Equation 4.12 defines the spirals problem in d dimensions, with $0 < r_0 < 1$ and $\omega > 0$..

$$\begin{cases} r = \sqrt{x_1(t)^2 + x_2(t)^2}, \\ x_1'(t) = -x_2(t) + \omega(r - r_0)x_1(t), \\ x_2'(t) = x_1(t) + \omega(r - r_0)x_2(t), \\ x_i'(t) = -\sigma x_i(t), i \in \{3, .., d\}, \end{cases} \quad \text{with } -1 \leq x_i(t) \leq 1, i \in \{1, .., d\}. \tag{4.12}$$

Qualitatively, in the plane P generated by canonical basis vectors $b_1 = (1, 0, ..., 0)$ and $b_2 = (0, 1, 0, ..., 0)$ and which includes point $c = (0, .., 0)$, the dynamics turns around c in P and if the distance to c is higher than r_0, then the trajectory is a spiral which increases its distance to c while the spiral goes towards c when the distance to c is smaller than r_0. When the point is exactly at the distance r_0 from c, the radius is kept constant. The parameter ω rules the increase or decrease of the distance to c. When x does not belong to P, the dynamics is the combination of the spiral in the plane generated by (b_1, b_2) which includes x and a translation towards P with a speed proportional to the distance from x to P (ruled by parameter σ). For sake of simplicity, there is no control in this problem (in our framework, the control set includes a single value).

In the spirals problem, $j^{\sharp}_{EDT}(\theta, K)$, the supremum of number of steps in K for the non-viable points of K, is infinite, while it is finite for the two first problems. Indeed, consider $x \in K$ such that $\|x - c\| = r_0 + \epsilon$ ($\epsilon > 0$); when ϵ tends to 0, the trajectory starting from x makes an indefinitely increasing number of turns around c before exiting from the constraint set. In the 2D example, we set $\omega = 0.01$ and the system makes a large number of rounds before exiting K even when it is moderately close to the circle of centre c and radius r_0.

It can be verified that the backstop viability kernel exists and is equal to the viability kernel. The viability kernel can easily be defined analytically:

$$Viab(K) = \{x \in K \,|\, x_1^2 + x_2^2 \leq r_0^2\}. \tag{4.13}$$

4.1.5. Parameter values. Table 1 breaks down the parameter values used in the tests of the population and consumption models and Table 2 provides the values used in the tests of the spirals model.

TABLE 1
Parameters of population and consumption models.

Model	m_1	M_1	m_2	M_2	m_u	M_u	α	β	δ	θ	j_A
Popul.2D	0.2	3	2	2	0.5	0.5	0	n.a.	0.1	0.6	50
Popul.dD	0.2	3	2	2	2	0.5	0.15	0.6	0.1	0.3	50
Cons.2D	0	2	0	2.5	0.9	0.9	0	n.a.	0.1	0.4	50
Cons.dD	0	2	0	2.2	0.9	0.9	-0.15	0.6	0.1	0.4	50

TABLE 2
Parameters of spirals model.

	r_0	ω	σ	δ	θ	j_A
2D	0.75	0.01	n.a.	0.1	0.6	10000
dD	0.75	0.1	0.2	0.1	0.6	100

In both tables the values are given for problems with modified axis scales so that $K = [0, 1]^d$. We checked experimentally that these parameters are such that

$Viab(K) = BSViab(K) = BSViab_{EDT}(\theta, \delta, K)$. Note that $m_u = 2$ in the Population model in d dimensions in order to get a smoother boundary $\partial V_+(x)$, ensuring that the conditions of the convergence for resistars are satisfied in the considered range of values of n.

4.2. Set approximation algorithms and their convergence rates. In this subsection, we assume $K = [0,1]^d$. The tests use two set approximation algorithms.

4.2.1. Nearest vertex. The nearest vertex approximation $N_n(H)$ of set $H \subset K$ is defined from G_n, the set of vertices of the regular grid covering K, as follows:

$$x \in N_n(H) \Longleftrightarrow \exists \xi \in G_n \cap H \mid \|x - \xi\| = \min_{\xi' \in G_n} \|x - \xi'\| . \tag{4.14}$$

PROPOSITION 4.1. *For any $H \subset K$ such that, for all $x \in H$ there exists $\xi \in G_n \cap H$ such that $D(x,\xi) \leq \frac{\sqrt{d}}{n}$, we have:*

$$D_h(H, N_n(H)) = \mathcal{O}(n^{-1}). \tag{4.15}$$

Proof. By definition, for all point $x \in K$ such that $x \in N_n(H)$ there exists $\xi \in (G_n \cap H)$ such that ξ is the point of G_n which is the nearest to x, thus $D(x,\xi) \leq \frac{\sqrt{d}}{2n}$. Therefore, $D(x,H) \leq \frac{\sqrt{d}}{2n}$. Moreover, by hypothesis, for all $x \in H$ there exists $\xi \in G_n$ such that $D(x,\xi) < \frac{\sqrt{d}}{n}$. Therefore, $D(x, N_n(H)) \leq \frac{\sqrt{d}}{n}$. We get: $D_h(H, N_n(H)) \leq \frac{\sqrt{d}}{n}$. ∎

4.2.2. Recursive simplex stars (resistars). We assume that set $H \subset K$ is a d-dimensional manifold, ∂H, the boundary of H and $\partial_K H = \partial H - (\partial H \cap \partial K)$ are $(d-1)$-dimensional manifolds. The resistars are designed for approximating $\partial_K H$ by hypersurfaces made of $(d-1)$-dimensional simplices. The first step for deriving resistar surfaces is determining the boundary points $B_H(G_n)$ which are approximations of the intersections between $\partial_K H$ and the edges of the grid. These points are computed on the edges $[v, v']$ of the grid such that one of the vertices is inside H and the other is outside. The estimation of a boundary point is done by successive dichotomies. The simplices defined in a cube share the barycentre of the boundary points located in the cube as a vertex, and their other vertices are defined similarly in the facets and faces of the cube. The complete description is available in [11].

THEOREM 4.2 ([11]). *If $\partial_K H$ is a $(d-1)$-dimensional manifold in K of reach[1] r such that $r > \sqrt{2}dn^{-1}$, if for all j-dimensional faces F of K, setting $H_F = H \cap F$, $\partial_F H_F$ is a $(j-1)$-dimensional manifold of reach $r_F > \sqrt{2}jn^{-1}$, and if all the boundary points are determined with at least $\log_2(n)$ dichotomies, then the Hausdorff distance between H and its resistar approximation decreases like $\mathcal{O}(dn^{-2})$.*

4.3. Evaluating the Hausdorff distance between the viability kernel and its approximations. The procedure evaluates tho distance from $\hat{H}^q$ (result of the approximation algorithm) to $Viab(K)$ (known analytically). It assumes that it is close to the distance from $Viab(K)$ to $\hat{H}^q$. It focuses on the points of the boundary $\partial_K \hat{H}^q$ that are likely to be the furthest to $\partial Viab(K)$ in order to limit the computation time.

[1] The reach of $\partial_K H$ is the supremum of ρ such that for any point x of K for which $D(x, \partial_K H) = \rho$, there is only one point $y \in \partial_K H$ such that $\|x - y\| = \rho$ [16].

4.3.1. Resistar approximation. Among the vertices of the simplices defining $\partial_K \hat{H}^q$, the barycentres of all boundary points of a grid cube are likely to be the furthest from $\partial_K Viab(K)$. The procedure estimates the distance from such vertices x to the projection of x on $\partial_K Viab(K)$ parallel to the estimated normal vector to the resistar surface at x and it returns the maximum of these distances. It is indeed assumed that the normal vector to $\partial_K \hat{H}^q$ at x is close to the normal vector to $\partial Viab(K)$ at the nearest point to x of $\partial Viab(K)$. The procedure estimates the intersection y of $\partial Viab(K)$ with the segment $[x, x + \nu D_M]$ or with the segment $[x, x - \nu D_M]$ (D_M being a parameter) by performing successive dichotomies. The case when $x + \nu D_M$ or $x - \nu D_M$ is located outside K requires a specific management. The estimation of the normal vector to $\partial_K \hat{H}^q$ at point x, is based on the estimation (by successive dichotomies) of $d - 1$ affinely independent points of $\partial_K \hat{H}^q$ at a given distance from x, from which the normal vector can be derived. The method includes a specific tratment for the cases when x is on the border of K or very close to it.

4.3.2. Nearest vertex approximation. The set $\hat{H}^q$ is now the nearest vertex approximation. The estimation of the Hausdorff distance from $\hat{H}^q$ to $Viab(K)$ is derived from the method defined for the resistar set approximation. It uses indeed the resistar surface denoted $H^\star$, defined from the boundary points $b = (v+v')/2$ where v and v' define a grid edge $[v, v']$ such that $v \in \hat{H}^q$ and $v' \notin \hat{H}^q$. It estimates the distance from the centres of the cubes containing boundary points of $H^\star$ to $\partial Viab(K)$ using the normal to $H^\star$ estimated at the barycentre of the boundary points of the cube. Indeed, the centre of the cube is always on the boundary of $\hat{H}^q$ and is likely to be the point from which $\partial Viab(K)$ is the furthest. The direction normal to $\partial_K H^\star$ at the barycentre of the boundary points of the cube is a reasonable approximation of the direction from the centre of the cube to its nearest point in $\partial Viab(K)$.

4.4. Results. Figure 1 shows the sets $\hat{H}^i$ for all the iterations of algorithm 3.1 applied on the 2D problems. The final result can be visually compared with the theoretical viability kernel (quantified evaluations of the Hausdorff distance are shown on figure 3). Note that the algorithm stops after 4 iterations for the population and consumption problems and after 6 iterations for the spirals problem (even with $\omega = 0.01$). The number of iterations is similar in higher dimensionality. Figure 2 shows examples of final results on the 3D problems. The smooth non-linearity along the x_2 axis ruled by parameters α and β appears in the viability kernel approximations of population and consumption problems. Panels (a), (b) and (c) of Figure 4 show the intersection with three chosen hyperplanes of viability kernel resistar approximations of algorithm 3.1 in 6 dimensions with $n = 5$ (grid size: 5^6).

On Figure 3, panels (a), (b) and (c) show the estimated Hausdorff distance between the viability kernel and its approximation (y axis) by the nearest vertex and by a resistar surface, in dimensionality 2, 3, 4 and 5, and for different values of the grid size n (x axis). For resistar approximations, the values of n are 7, 9, 13, 17, 25, 33, 49, 65, 97, 129, 193, 257 for the 2D problems, 7, 9, 13, 17, 25, 33, 49, 65, 97 for the 3D problems, 7, 9, 13, 17, 25, 33, 49 for the 4D problems and 7, 9, 13, 17, 25 for the 5D problems. For nearest vertex approximations, the values 7 and 9 are not tested because they are too small for defining properly the sets B^i_{in}. The other values of n are the same The axes are in a logarithmic scale.

Table 3 shows the estimation of the slopes of the logarithm of the Hausdorff distance as a function of the logarithm of n, for the problems in 2 and 3 dimensions. These results are in good agreement with the theoretical prediction of an Hausdorff distance decreasing like n^{-1} for the approximation with the nearest vertex and like

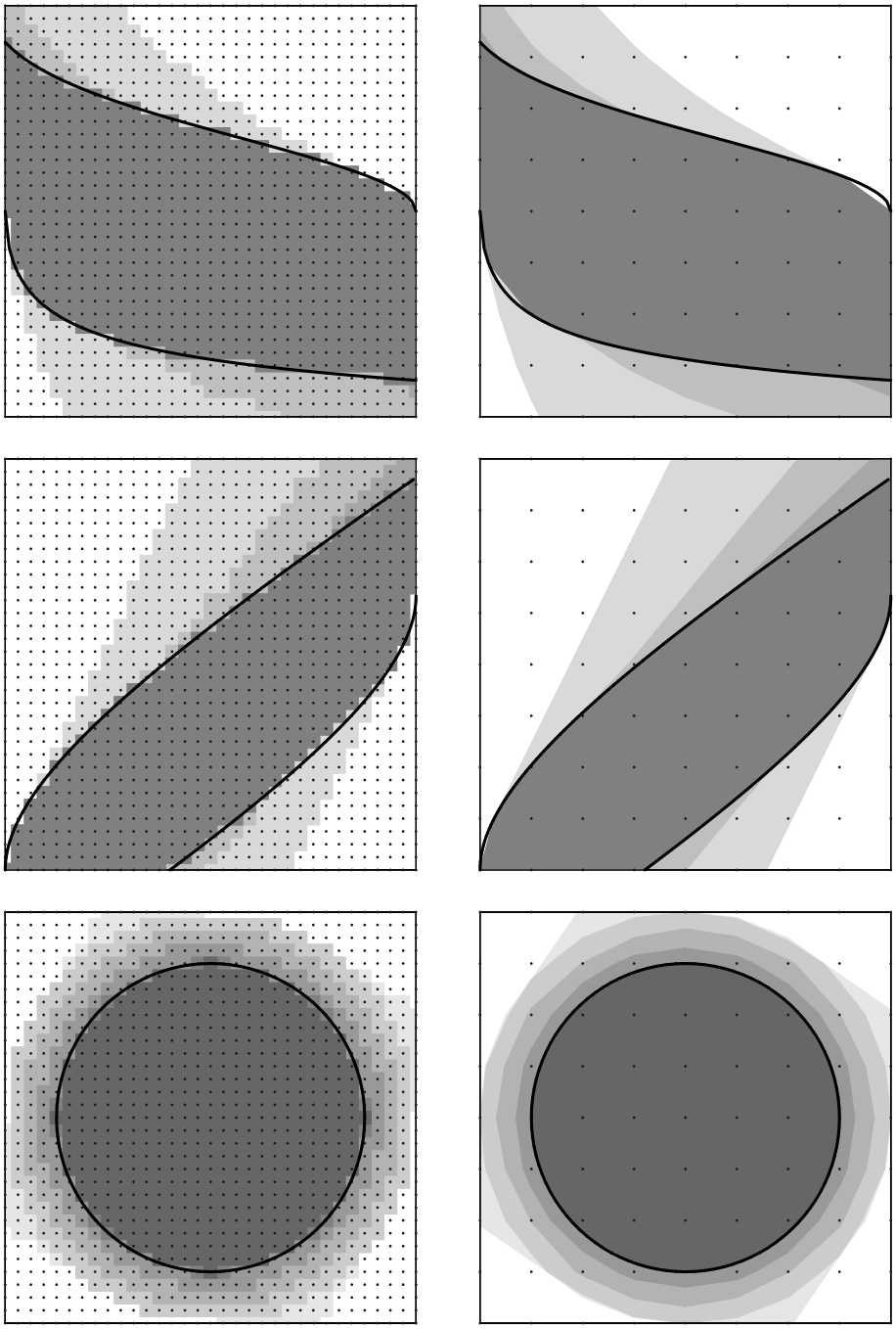

FIGURE 1. *Approximations of the viability kernel in 2D for the three problems (first row: population, second row: consumption, third row: spirals). Left column: nearest vertex approximation with $n = 33$ (grid size: 33^2), right column: resistar approximation with $n = 9$ (grid size: 9^2). The black curves are the boundaries of the theoretical viability kernel. The approximations $\hat{H}^i$ are represented in darker and darker grey as i increases. The darkest set is the output of algorithm 3.1.*

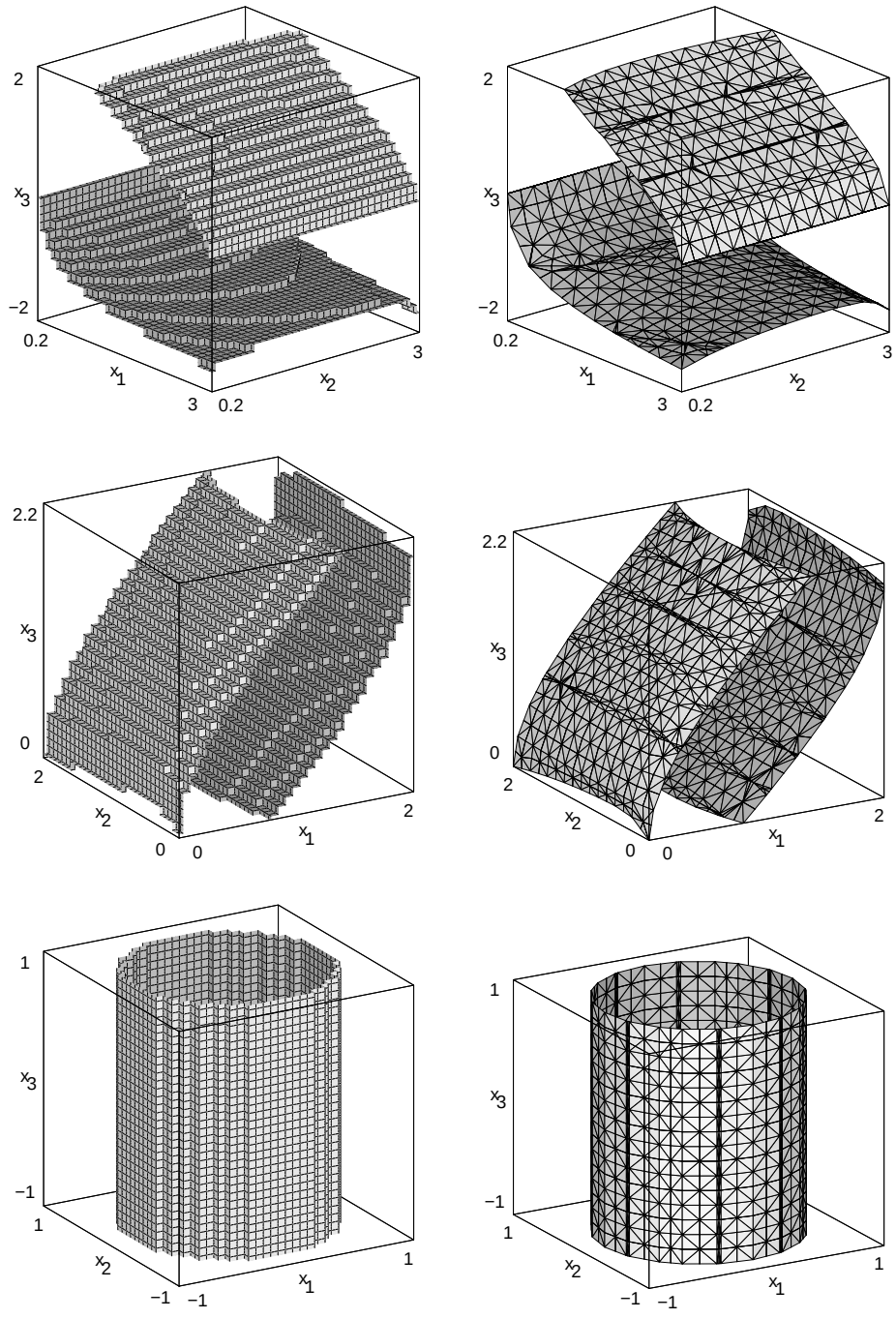

FIGURE 2. *Approximations of the viability kernel in 3D for the three problems (first row: population, second row: consumption, third row: spirals). Left column: nearest vertex approximation with $n = 33$ (grid size: 33^3), right column: resistar approximation with $n = 9$ (grid size: 9^3).*

	Nearest v.		Resistars	
	2D	3D	2D	3D
Population	-1.00	-0.97	-2.00	-1.99
Consumption	-1.01	-1.03	-2.01	-1.98
Spiral	-1.02	-0.94	-2.02	-2.01

TABLE 3

Slopes of the linear regression of the logarithm of the estimated Haussdorf distance between the viability kernel and its approximation as a function of the logarithm of n. The R^2 values are all superior to 0.98.

n^{-2} for the approximation with the resistars.

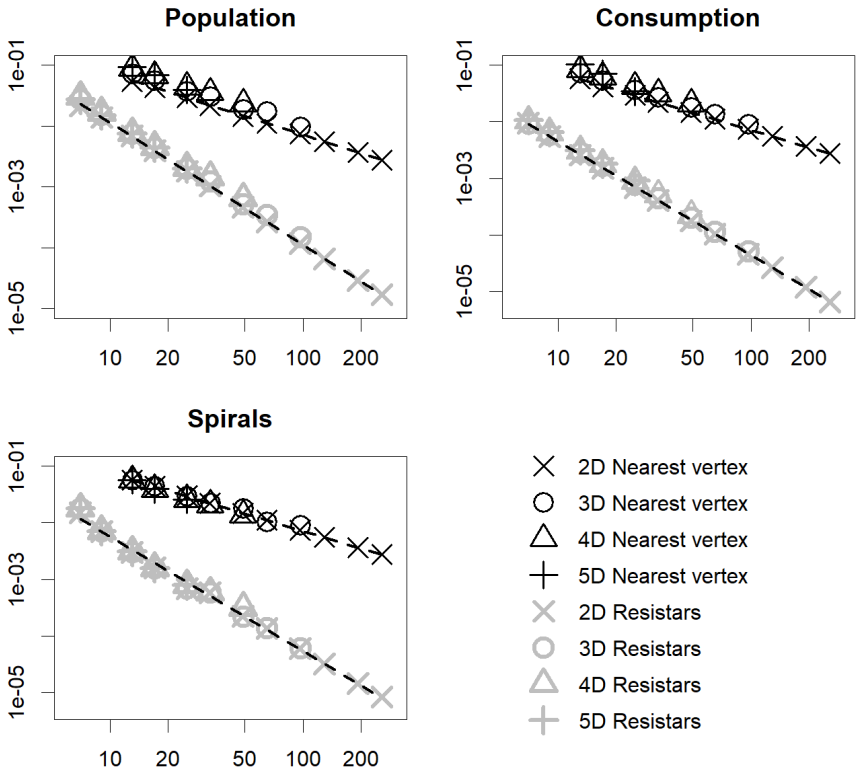

FIGURE 3. *Estimation of the Hausdorff distance between the viability kernel and its approximation (y axis) as a function of n defining the grid size as n^d (x axis) for the population, consumption and spirals problems in $d \in \{2, 3, 4, 5\}$ dimensions. The dashed lines are the linear regressions on the 2D values (slopes given in table 3).*

On figure 3, for a given value of n, the error does not change much when increasing the dimensionality. This observation is confirmed on Figure 4, panel (d) showing that the error of the resistar approximation for a grid defined by $n = 5$ does not vary significantly when the dimensionality d varies from 3 to 9.

5. Discussion - conclusion. In general conditions, there exists a range of parameter values for which `BSViabApp`(n, δ, θ, j_A), the output of the approximation

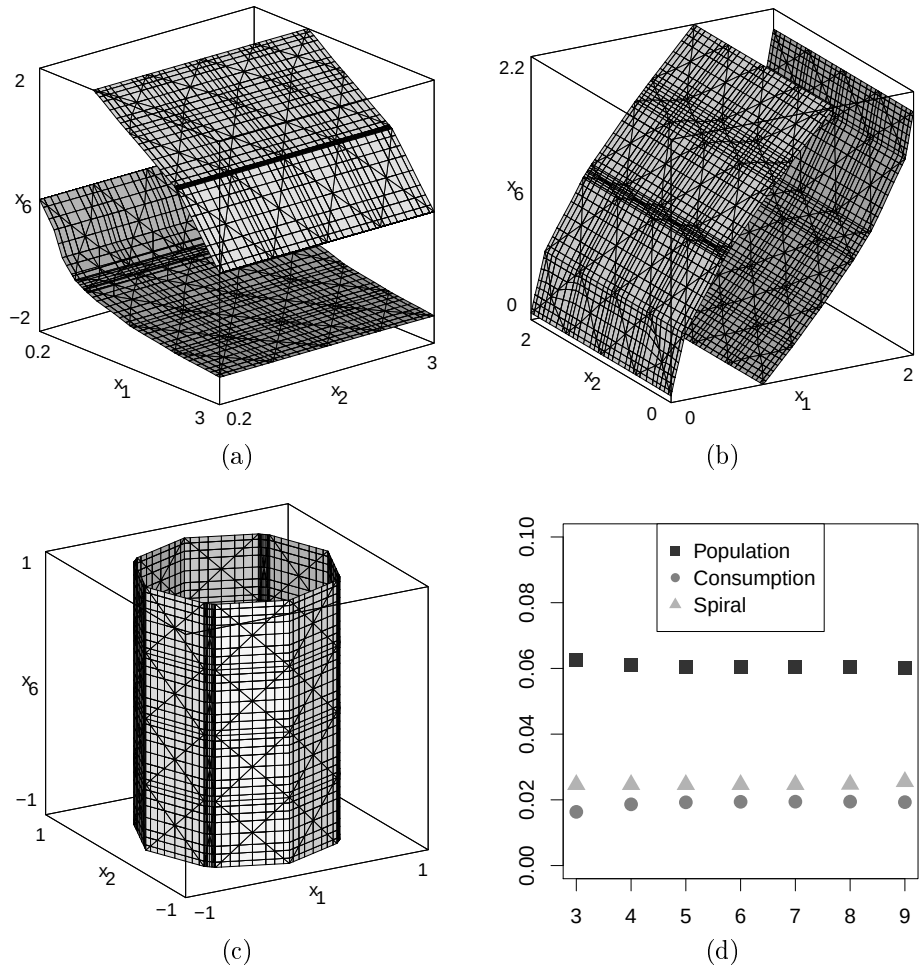

FIGURE 4. *Panels (a), (b) and (c): Resistar backstop viability kernel approximations in dimensionality $d = 6$ and $n = 5$ (grid size: 5^6), for respectively the population, consumption and spirals problems. Each panel represents the intersection of the resistar approximation with 3 hyperplanes ($x_3 = 0.02$, $x_4 = 0.05$ and $x_5 = 0.08$ in $K = [0,1]^6$). Panel (d): Hausdorff distance between viability kernel and its resistar approximation (y axis) for the population, consumption and spirals problems in a grid of size $n = 5$ and dimensionality varying from 3 to 9 (x axis).*

algorithm, satisfies:

$$BSViab(\frac{3\delta}{2}, \theta, K) \cap \Gamma(n) \subset \texttt{BSViabApp}(n, \delta, \theta, j_A) \cap \Gamma(n) \subset Viab_{EDT}(\theta, K) \cap \Gamma(n).$$

The second inclusion implies that all the points of $\texttt{BSViabApp}(n, \delta, \theta, j_A) \cap \Gamma(n)$ are viable. This is generally not guaranteed with the Saint-Pierre algorithm which provides an approximation of a set which contains the viability kernel. This difference is important when the objective is to guarantee the viability of a system. When the backstop viability kernel $BSViab(K)$ exists, under general conditions, there exists a range of parameter values such that:

$$\texttt{BSViabApp}(n, \delta, \theta, j_A) \cap \Gamma(n) = BSViab(K) \cap \Gamma(n).$$

Overall, these results lead to a convergence rate of the algorithm to the backstop viability kernel which is the same as the convergence rate of the chosen set approximation technique.

When the viability kernel is equal to the backstop viability kernel, the conditions to get this convergence rate are more general. Even when using the nearest vertex approximation, we expect our algorithm to then outperform the current techniques approximating viability kernels for three reasons. Firstly, the convergence to the viability kernel is ensured without decreasing the time step to 0, which is a major difference. Secondly, for a given time step, our algorithm requires a lower number of iterations, especially when the supremum of time steps in K for the non-viable points is infinite (as illustrated on the spiral problem). Thirdly, our algorithm avoids cumulating the error of the successive set approximations taking place in standard algorithms.

When using resistars as set approximation technique, if the best conditions are satisfied, the convergence rate of our algorithm is like $\mathcal{O}(n^{-2})$ which significantly increases the advantage over the standard methods, converging at best like $\mathcal{O}(n^{-1})$. Indeed, in order to be as accurate as a resistar approximation using a grid of n^d points, the standard methods need a grid of at least n^{2d} points. For instance, we have shown that it is possible to run resistars approximations in 5 dimensions using a grid of 25^5 (about $8\ 10^6$) points. In order to reach the same accuracy as the one of these approximations, the standard methods would require a grid of at least $(25^2)^5$ (about $9.5\ 10^{13}$) points, which is not manageable by current standard computers.

REFERENCES

[1] I. Alvarez, R. Reuillon, and R. D. Aldama, *Viabilititree: a kd-tree framework for viability-based decision*, archives-ouvertes.fr, (2016).

[2] J. Aubin, *Viability theory*, Birkhäuser, 1991.

[3] J. Aubin, *Elements of viability theory for the analysis of dynamic economics*, Ecole thematique du CNRS 'Economie Cognitive', (2002).

[4] J.-P. Aubin, A. Bayen, and P. Saint-Pierre, *Viability Theory: New Directions*, Springer, 2011.

[5] O. Bokanowski, S. Martin, R. Munos, and H. Zidani, *An anti-diffusive scheme for viability problems*, Applied Numerical Mathematics, 56 (2006), pp. 1147–1162.

[6] N. Bonneuil, *Computing the viability kernel in large state dimension*, Journal of Mathematical Analysis and Applications, 323 (2006), pp. 1444–1454.

[7] P. Cardaliaguet, M. Quincampois, and P. Saint-Pierre, *Numerical schemes for discontinuous value functions of optimal control*, Set-Valued Analysis, 8 (2000), pp. 11–126.

[8] L. Chapel, X. Castello, C. Bernard, G. Deffuant, V. Egiluz, and S. Martin, *Viability and resilience of languages in competition*, PLOS ONE, 5 (2010).

[9] L. Chapel, G. Deffuant, S. Martin, and C. Mullon, *Defining yield policies in a viability approach*, Ecological Modelling, 212 (2008), pp. 10–15.

[10] P. Coquelin, S. Martin, and R. Munos, *A dynamic programming approach to viability problems*, in IEEE ADPRL Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007.

[11] G. Deffuant, *Recursive simplex stars*, arxiv.org/abs/1707.08373, (2018), http://arxiv.org/abs/1707.08373.

[12] G. Deffuant, L. Chapel, and S. Martin, *Approximating viability kernel with support vector machines*, IEEE Transactions on Automatic Control, 52 (2007), pp. 933–937.

[13] G. Deffuant and N. Gilbert, eds., *Viability and Resilience of Complex Systems: Concepts, Methods and Case Studies from Ecology and Society*, Springer, 2011.

[14] M. Delara and L. Doyen, *Sustainable Management of Natural Resources. Mathematical Models and Methods*, Springer, 2008.

[15] L. Doyen, A. Cissé, S. Gourget, L. Mouysset, P. Hardy, C. B. B. . Blanchard, F. Jiguet, J. Perreau, and O. Thébaud, *Ecological-economic modelling for the sustainable management of biodiversity*, Computational Management Science, 10 (2013), pp. 353–

364.
[16] H. Federer, *Curvature measures*, Tansactions of the American Mathematical Society, 93 (1959), pp. 418–491.
[17] J. Heitzig, T. Kittel, J. F. Donges, and N. Molkenthin, *Topology of sustainable management of dynamical systems with desirable states: from defining planetary boundaries to safe operating spaces in the Earth system*, Earth System Dynamics, 7 (2016), pp. 21–50, https://doi.org/10.5194/esd-7-21-2016.
[18] M. Kalisiak and M. van de Panne, *Approximate safety enforcement using computed viability envelopes*, in Proceedings of IEEE International Conference on Robotics and Automation, 2004, pp. 4289–4294.
[19] J. B. Krawczyck and A. Pharo, *Viability Kernel Approximation, Analysis and Simulation Application - VIKAASAViability Kernel Approximation, Analysis and Simulation Application - VIKAASA*, 2011.
[20] J. Maidens, S. Kaynama, I. Mitchell, M. Oishi, and G. Dumont, *Lagrangian methods for approximating the viability kernel in high-dimensional systems*, Automatica, 49 (2013), pp. 2017–2029.
[21] S. Martin, *The cost of restoration as a way of defining resilience: a viability approach applied to a model of lake eutrophication*, Ecology and Society, 9(2) (2004), http://www.ecologyandsociety.org/vol9/iss2/art8.
[22] J. Mathias, B. Bonté, T. Cordonnier, and F. DeMorogues, *Using the viability theory for assessing flexibility of forest managers under ecological intensification*, Environmental Management, 56 (2015), pp. 1170–1183.
[23] S. Mesmoudi, I. Alvarez, S. Martin, R. Reuillon, M. Sicard, and N. Perrot., *Coupling geometric analysis and viability theory for system exploration: Application to a living food system*, Journal of Process Control, (2014).
[24] A. Oubraham and G. Zaccour, *A survey of applications of viability theory to the sustainable exploitation of renewable resources*, Ecological economics, 145 (2018), pp. 346–367.
[25] J. Rieger, *Shadowing and the viability kernel algorithm*, Applied Mathematics and Optimization, 60 (2009), pp. 429–441.
[26] C. Rougé, J.-D. Mathias, and G. Deffuant, *Extending the viability theory framework of resilience to uncertain dynamics, and application to lake eutrophication*, Ecological Indicators, 29 (2013), pp. 420–433.
[27] R. Sabatier, L. Doyen, and M. Tichit, *Action versus result-oriented schemes in a grassland agroecosystem: a dynamic modelling approach*, PLOS ONE, 7 (2012).
[28] P. Saint-Pierre, *Approximation of viability kernel*, App. Math. Optim., 29 (1994), pp. 187–209.
[29] A. Schulbauer and R. Sumala, *Economic viability and small-scale fisheries: A review*, Ecological Economics, 124 (2016), pp. 69–75.
[30] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, *Computational techniques for the verification and control of hybrid systems*, in Proceedings of IEEE 91, 2003.