Eulerian state estimation

Thomas Le Mézo, Luc Jaulin and Benoît Zerr
{thomas.le_mezo,luc.jaulin,benoit.zerr}@ensta-bretagne.fr
ENSTA-Bretagne, LabSTICC, CNRS 6285,
2 rue François Verny, 29806 Brest, France.

June 2, 2017

Abstract

In this paper, we present a tool for computing an inner and an outer approximations of the largest positive invariant set associated with a nonlinear state equation. Further, we show how to solve rigorously complex problems related to continuous time dynamical systems, such as the Eulerian state estimation problem.

1 Introduction

Invariant sets are used in nonlinear control theory [2] [3], for instance to validate *(i)* some properties of cyber-physic systems [10][27], *(ii)* to ensure the safe take off [25] of an airplane or *(iii)* to avoid collisions [6] with other aircrafts. In this paper, we deal with a dynamical system $\mathcal{S}$ defined by the following state equation:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the evolution function of $\mathcal{S}$ [7, 8]. Denote by $\boldsymbol{\varphi}$ the flow map of $\mathcal{S}$, *i.e.*, with the initial condition $\mathbf{x}_0 = \mathbf{x}(0)$, the system $\mathcal{S}$ reaches the state $\boldsymbol{\varphi}(t, \mathbf{x}_0)$ at time t.

Two different types of approaches [16] are used to deal with the estimation of the solution for (1): the Lagrangian and the Eulerian. This classification is taken from the field of fluid mechanics [11]. In the Lagrangian point of view, the observer follows an individual fluid parcel as it moves through the fluid. In an Eulerian point of view, the observer stays at the same place and looks at fluid motion moving around him.

When we deal with a dynamical system such as (1), the speed of the fluid corresponds to the evolution function $\mathbf{f}(\mathbf{x}(t))$ and the position of a fluid parcel at time t corresponds to the state $\mathbf{x}(t)$. A Lagrangian approach would require simulations to find states that reach the target [20]. In the literature, this method is generally restricted to linear dynamics [1] where a closed form for the flow $\boldsymbol{\varphi}$ is available. It can also be used for nonlinear systems if we use guaranteed integration [24] [28], but the resulting method is slow. As shown in [12] [17] [9] a Lagrangian method requires many bisections with respect to the time line (for the integration of the state equation), but also on the state space. The Eulerian methods are used for nonlinear systems [19] and try to avoid the integration of the state equation but the corresponding algorithms rely on gridding the state space [22]. To provide guaranteed results, gridding methods require the knowledge of some Lipschitz constant which are rarely available in practice [23]. Lyapunov-based methods [21][5], level-set methods [15], or barrier functions [4] can also be considered as Eulerian since they only check the constraints on the state space and do not need to perform any integration though time. Now, these methods required a parametric expression for candidate Lyapunov-like functions [26].

This paper deal with Eulerian state estimation which can be formalized as follows:

$$\begin{array}{lcc} (i) & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) & \text{(evolution)} \\ (ii) & \mathbf{x}(t_i) \in \mathbb{X}_i & \text{(event)} \\ (iii) & \forall\,(i,j) \in \mathbb{J}, t_i \leq t_j & \text{(precedence)} \end{array} \tag{2}$$

where the t_i are unknown times. Equation (i) corresponds to (1). Constraint *(ii)* tells us that at some unknown times t_i the trajectory has crosses a known set $\mathbb{X}_i$. This set corresponds to an untemporal observation such as: "the robot entered in my house". We deduce this information from the existence of wheelprints, for instance, but we do not know when this event occurred. Constraint (iii) expresses an order between event and is represented by a set $\mathbb{J} \subset \mathbb{N}^2$. For instance, if $(2,5) \in \mathbb{J}$, then the trajectory has crossed $\mathbb{X}_2$ before $\mathbb{X}_5$. Equations (ii) and (iii) can be represented graphically by a graph or a Petri net [18]. Solving such an estimation problem amounts to finding all states that are consistent with one trajectory satisfying (2).

This paper presents an approach based on invariant sets to solve this problem.

2 Invariant sets

This section presents some definitions on invariant sets. We also show that the solution set of several problems involving dynamical systems can be expressed as an algebraic expression involving maximal positive invariant sets.

Positive invariant set. A set $\mathbb{A}$ is *positive invariant* [3] if for any trajectory $\mathbf{x}(\cdot)$ of (1), we have

$$\mathbf{x}(0) \in \mathbb{A}, t \geq 0 \implies \mathbf{x}(t) \in \mathbb{A}. \tag{3}$$

The set of all positive invariant sets is a lattice, *i.e.*, the union and the intersection of two positive invariant sets is positive invariant. A consequence is that, given a set $\mathbb{X}$, the notion of *largest positive invariant set* contained in $\mathbb{X}$ and *smallest positive invariant set* enclosing $\mathbb{X}$ can be defined.

Largest positive invariant set. Given a set $\mathbb{X}$, there exists a largest

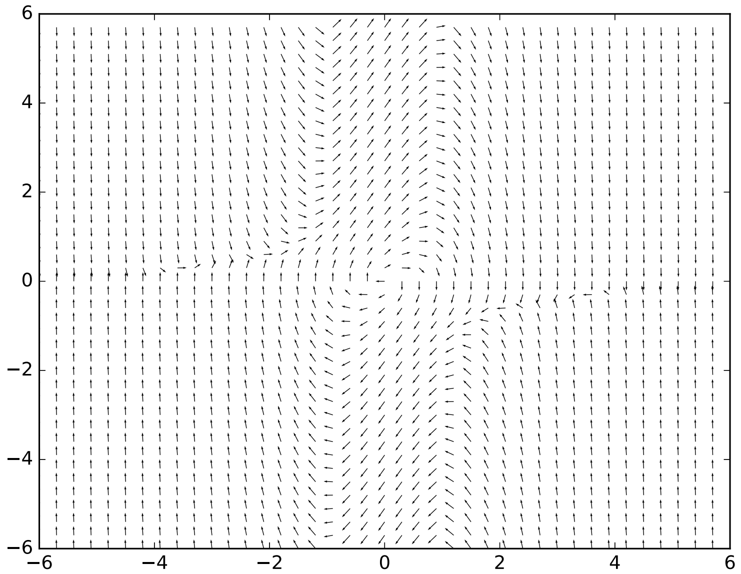

Figure 1: Vector field associated to the Van der Pol system in the box $[-6, 6]^{\times 2}$

positive invariant set for (1) included in $\mathbb{X}$. It is given by

$$Inv^{+}(\mathbf{f}, \mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \geq 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\}. \tag{4}$$

In [14][13] a tool has been developed to compute an inner and an outer approximation for $Inv^{+}(\mathbf{f}, \mathbb{X})$. As an illustration, consider the system described by the Van der Pol equation:

$$\begin{cases} \dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & \left(1 - x_1^2\right) \cdot x_2 - x_1. \end{cases} \tag{5}$$

Figure 1 provides an illustration of its vector field.

Figure 2 shows that the largest positive invariant set in $\mathbb{X} = [-6, 6]^{\times 2}$ associated to (5). All points in the magenta area will stay inside $\mathbb{X}$ forever whereas all points in the blue zone will leave $\mathbb{X}$.

Largest negative invariant set. It corresponds to the set

$$Inv^{-}(\mathbf{f}, \mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \leq 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\}. \tag{6}$$

Since

$$Inv^{-}(\mathbf{f}, \mathbb{X}) = Inv^{+}(-\mathbf{f}, \mathbb{X}), \tag{7}$$

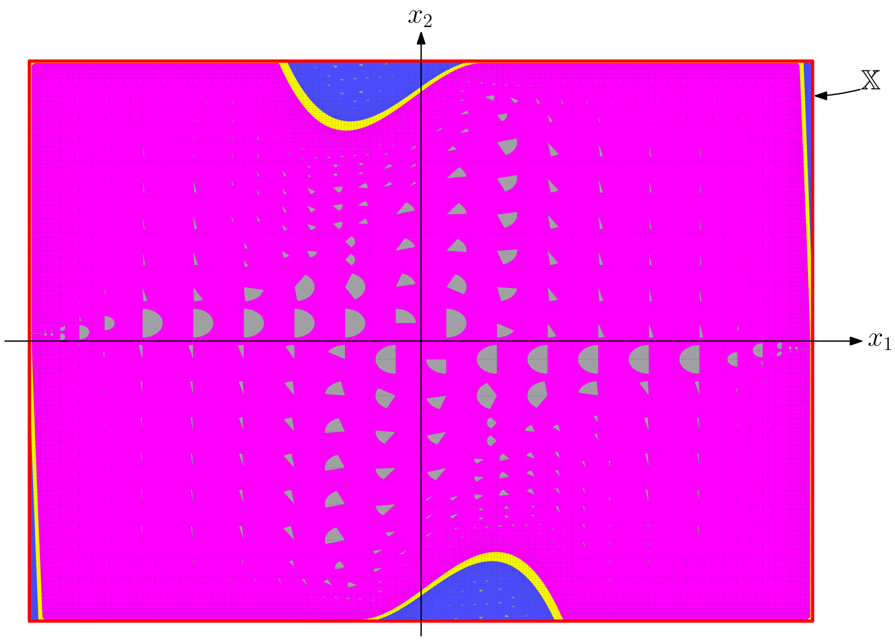

Figure 2: Largest positive invariant set $Inv^+(\mathbf{f}, \mathbb{X})$ where $\mathbb{X} = [-6, 6]^{\times 2}$ and $\mathbf{f}$ is the evolution function of the Van der Pol system

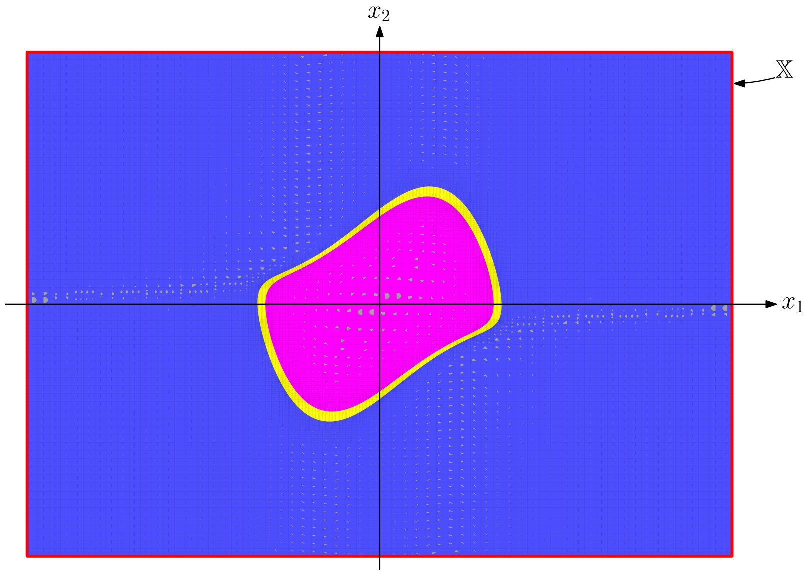

Figure 3: Largest negative invariant set $Inv^-(\mathbf{f}, \mathbb{X})$ where $\mathbb{X} = [-6, 6]^{\times 2}$ corresponding the Van der Pol system

the largest negative invariant set can be defined in terms of positive invariant sets. Figure 3 shows that the largest negative invariant set in $\mathbb{X} = [-6, 6]^{\times 2}$ associated to (5). All points in the magenta area will go to $\mathbb{X}$ in the future whereas all points in the blue zone will never reach $\mathbb{X}$.

Largest invariant set. It corresponds to the set

$$Inv(\mathbf{f}, \mathbb{X}) = \{\mathbf{x}_0 \mid \forall t \in \mathbb{R}, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\}. \tag{8}$$

We have

$$\begin{array}{rcl} Inv(\mathbf{f}, \mathbb{X}) & = & \{\mathbf{x}_0 \mid \forall t \le 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X} \wedge \forall t \ge 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\} \\ & = & \{\mathbf{x}_0 \mid \forall t \le 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\} \cap \{\mathbf{x}_0 \mid \forall t \ge 0, \boldsymbol{\varphi}(t, \mathbf{x}_0) \in \mathbb{X}\} \\ & = & Inv^+(-\mathbf{f}, \mathbb{X}) \cap Inv^+(\mathbf{f}, \mathbb{X}). \end{array} \tag{9}$$

Thus $Inv(\mathbf{f}, \mathbb{X})$ can be defined in terms of largest positive invariant sets. Since, in our Van der Pol example $Inv^-(\mathbf{f}, \mathbb{X}) \subset Inv^+(\mathbf{f}, \mathbb{X})$, the largest invariant set also corresponds to Figure 3.

Forward reach set. It corresponds to the set

$$Forw(\mathbf{f}, \mathbb{X}) = \{\mathbf{x} \mid \exists t \ge 0, \exists \mathbf{x}_0 \in \mathbb{X}, \boldsymbol{\varphi}(t, \mathbf{x}_0) = \mathbf{x}\}. \tag{10}$$

Since

$$\begin{array}{rcl} Forw(\mathbf{f}, \mathbb{X}) & = & \{\mathbf{x} \mid \exists t \ge 0, \exists \mathbf{x}_0 \in \mathbb{X}, \boldsymbol{\varphi}(-t, \mathbf{x}) = \mathbf{x}_0\} \\ & = & \{\mathbf{x} \mid \exists t \ge 0, \boldsymbol{\varphi}(-t, \mathbf{x}) \in \mathbb{X}\} \\ & = & \overline{\{\mathbf{x} \mid \forall t \ge 0, \boldsymbol{\varphi}(-t, \mathbf{x}) \in \overline{\mathbb{X}}\}} \\ & = & \overline{Inv^+(-\mathbf{f}, \overline{\mathbb{X}})} \end{array} \tag{11}$$

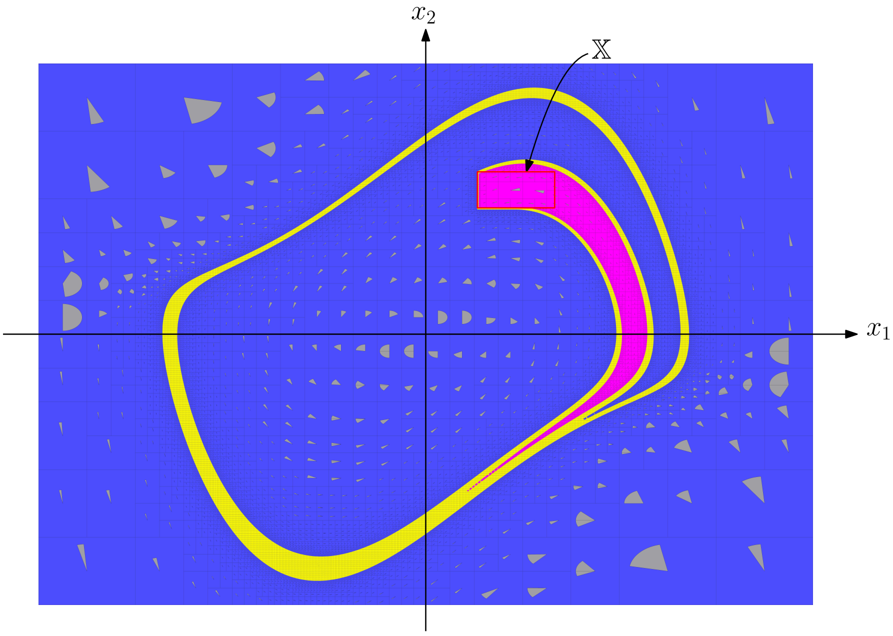

Figure 4: Forward reach set of the Van der Pol system. The frame box is $[-3,3]^{\times 2}$

the set $Forw(\mathbf{f},\mathbb{X})$ can be defined in terms of positive invariant sets. The set $Forw(\mathbf{f},\mathbb{X})$ corresponds to the smallest positive invariant set enclosing $\mathbb{X}=[0.4,1.0]\times[1.4,1.8]$. An illustration is given by Figure 4.

Backward reach set. It corresponds to the set

$$Back(\mathbf{f},\mathbb{X})=\{\mathbf{x}_0 \mid \exists t\geq 0, \boldsymbol{\varphi}(t,\mathbf{x}_0)\in\mathbb{X}\}\,. \tag{12}$$

Since

$$\begin{array}{rcl} Back(\mathbf{f},\mathbb{X}) & = & Forw(-\mathbf{f},\mathbb{X}) \\ & = & \overline{Inv^{+}(\mathbf{f},\overline{\mathbb{X}})} \end{array} \tag{13}$$

the set $Back(\mathbf{f},\mathbb{X})$ can be defined in terms of positive invariant sets. An illustration is given on Figure 5 for $\mathbb{X}=[0.4,1.0]\times[1.4,1.8]$. All points in the magenta area will reach $\mathbb{X}$ for some $t\geq 0$.

3 Eulerian state estimation

Define ℓ sets $\mathbb{X}_0,\mathbb{X}_1,\ldots,\mathbb{X}_\ell$ of the state space. Define $\mathbb{Z}_k^{forw}$ the set of all state vectors $\mathbf{x}(t)$ inside $\mathbb{X}_k$ that have visited $\mathbb{X}_0,\mathbb{X}_1,\ldots,\mathbb{X}_{k-1}$ in the past (*i.e.*, before time t) and in the specified order. We have

$$\mathbb{Z}_{k+1}^{forw}=Forw\left(\mathbb{Z}_k^{forw}\right)\cap\mathbb{X}_{k+1}, \tag{14}$$

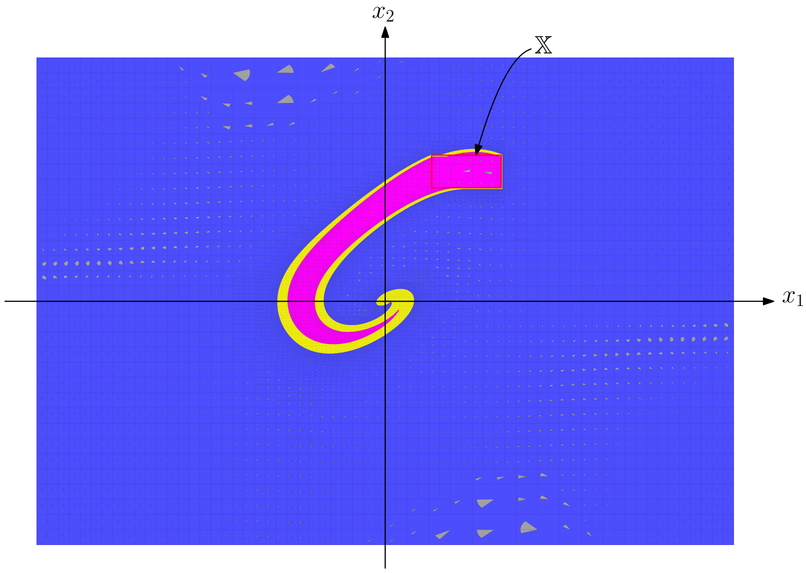

Figure 5: Backward reach set of the Van der Pol system. The frame box is $[-3,3]^{\times 2}$

with $\mathbb{Z}_0^{forw} = \mathbb{X}_0$. This sequence corresponds to what we call the *Eulerian filter*. The principle is illustrated by Figures 6,7 8 and 9. For simplicity, $Forw\left(\mathbf{f}, \mathbb{X}\right)$ and $Back\left(\mathbf{f}, \mathbb{X}\right)$ are denoted by $Forw\left(\mathbb{X}\right)$ and $Back\left(\mathbb{X}\right)$.

Define the set $\mathbb{Z}_k^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{X}_k$ that have visited $\mathbb{X}_0, \mathbb{X}_1, \dots, \mathbb{X}_{k-1}$ in the past and will visit $\mathbb{X}_{k+1}, \dots, \mathbb{X}_\ell$ in the future. We have

$$\mathbb{Z}_k^{back} = Back\left(\mathbb{Z}_{k+1}^{back}\right) \cap \mathbb{Z}_k^{forw}, \tag{15}$$

with $\mathbb{Z}_\ell^{back} = \mathbb{Z}_\ell^{forw}$. The will be called the *Eulerian smoother*. The Eulerian smoother is illustrated by Figures 1011

As illustrated by Figure 12, the set of trajectories that started inside $\mathbb{X}_0$ and visited the sets $\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_{\ell-1}$sequentially, and that ended in $\mathbb{X}_\ell$ can thus be enclosed by

$$Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_\ell^{back}\right). \tag{16}$$

Example 1. Define three sets $\mathbb{X}_0, \mathbb{X}_1, \mathbb{X}_2$ and assume that we want to find the set of trajectories that started inside $\mathbb{X}_0$ and visited the set $\mathbb{X}_1$ and then finally reached the set $\mathbb{X}_2$. This problem corresponds to an Eulerian state estimation problem where $\mathbb{J} = \{(0,1), (1,2)\}$, which means that $t_0 \leq t_1 \leq t_2$. We

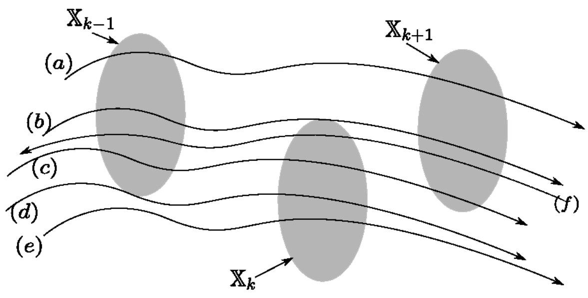

Figure 6: The trajectories (b),(c),(d) are consistent with the sets $\mathbb{X}_{k-1}, \mathbb{X}_k, \mathbb{X}_{k+1}$. The trajectory (f) does not respect the required order.

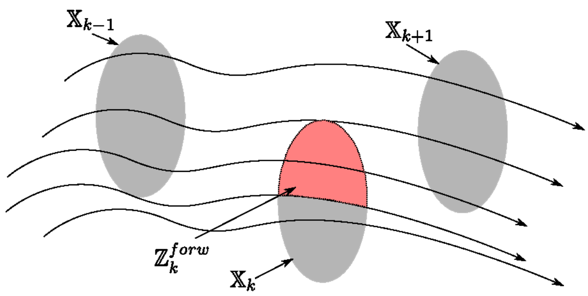

Figure 7: Set $\mathbb{Z}_k^{forw}$ of all states $\mathbf{x}(t)$ in $\mathbb{X}_k$ that have already visited $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{k-1}$

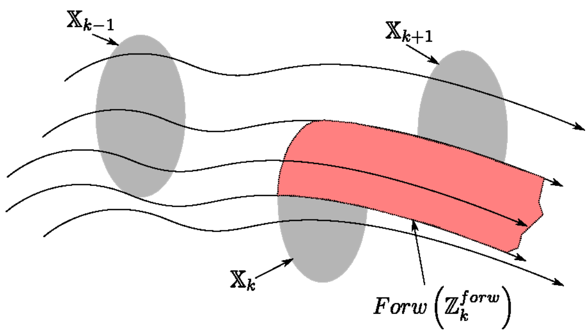

Figure 8: The set $Forw\left(\mathbb{Z}_k^{forw}\right)$ corresponds to all states $\mathbf{x}(t)$ that have visited $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_k$

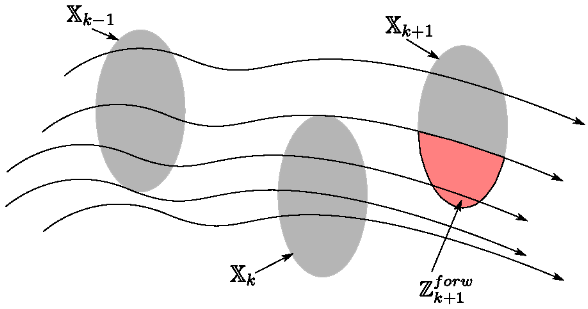

Figure 9: Set $\mathbb{Z}_{k+1}^{forw}$ of all states $\mathbf{x}(t)$ in $\mathbb{X}_{k+1}$ that have already visited $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_k$

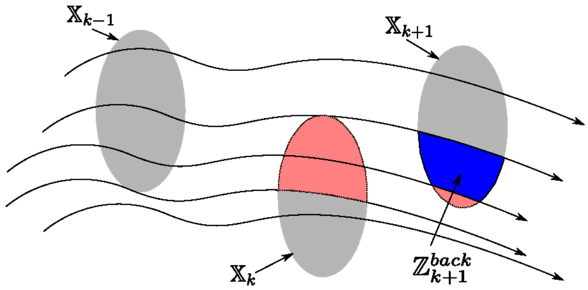

Figure 10: Set $\mathbb{Z}_{k+1}^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{Z}_{k+1}^{forw}$ that will visit $\mathbb{X}_{k+2}, \ldots, \mathbb{X}_\ell$

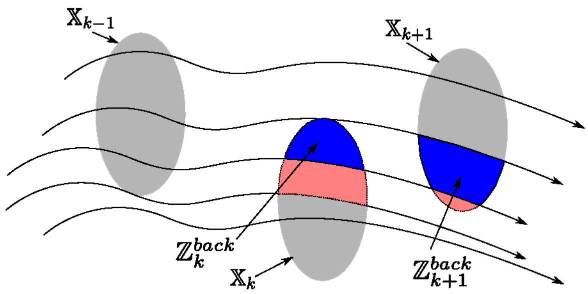

Figure 11: Set $\mathbb{Z}_k^{back}$ of all states $\mathbf{x}(t)$ inside $\mathbb{Z}_k^{forw}$ that will visit $\mathbb{X}_{k+1}, \ldots, \mathbb{X}_\ell$

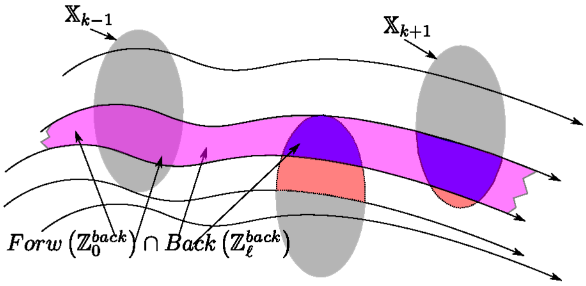

Figure 12: Set $Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_\ell^{back}\right)$ enclosing the trajectory consistent with the past and future visits

have

$$\begin{array}{ccc}
\mathbb{X} & = & Forw\left(\mathbb{Z}_0^{back}\right) \cap Back\left(\mathbb{Z}_2^{back}\right) \\
\mathbb{Z}_0^{forw} & = & \mathbb{X}_0 \\
\mathbb{Z}_1^{forw} & = & Forw\left(\mathbb{Z}_0^{forw}\right) \cap \mathbb{X}_1 \\
\mathbb{Z}_2^{forw} & = & Forw\left(\mathbb{Z}_1^{forw}\right) \cap \mathbb{X}_2 \\
\mathbb{Z}_2^{back} & = & \mathbb{Z}_2^{forw} \\
\mathbb{Z}_1^{back} & = & Back\left(\mathbb{Z}_2^{back}\right) \cap \mathbb{Z}_1^{forw} \\
\mathbb{Z}_0^{back} & = & Back\left(\mathbb{Z}_1^{back}\right) \cap \mathbb{Z}_0^{forw}
\end{array}$$

In the special case where $\mathbb{X}_0 = [\mathbf{a}] = [0, 0.6] \times [0.8, 1.8]$,$\mathbb{X}_1 = [\mathbf{b}] = [0.7, 1.5] \times [-0.2, 0.2]$ and $\mathbb{X}_2 = [\mathbf{c}] = [0.2, 0.6] \times [-2.2, -1.5]$, we get the set depicted on Figure 13.

An application of Eulerian state estimation is illustrated by Figure 14, where ocean currents are represented. The objective of the mission is to find a trajectory for a buoy that follows the currents in order to visit the three red boxes. Since the time is not taken into account in the estimation process, we are in an Eulerian context.

References

[1] E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Hybrid Systems: Computation and Control*, pages 20–35. Springer Berlin Heidelberg, 2003.

[2] J.P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhäuser, Boston, 1990.

[3] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control*. Springer Science & Business Media, October 2007.

[4] O. Bouissou, A. Chapoutot, A. Djaballah, and M. Kieffer. Computation of parametric barrier functions for dynamical systems using interval analysis.

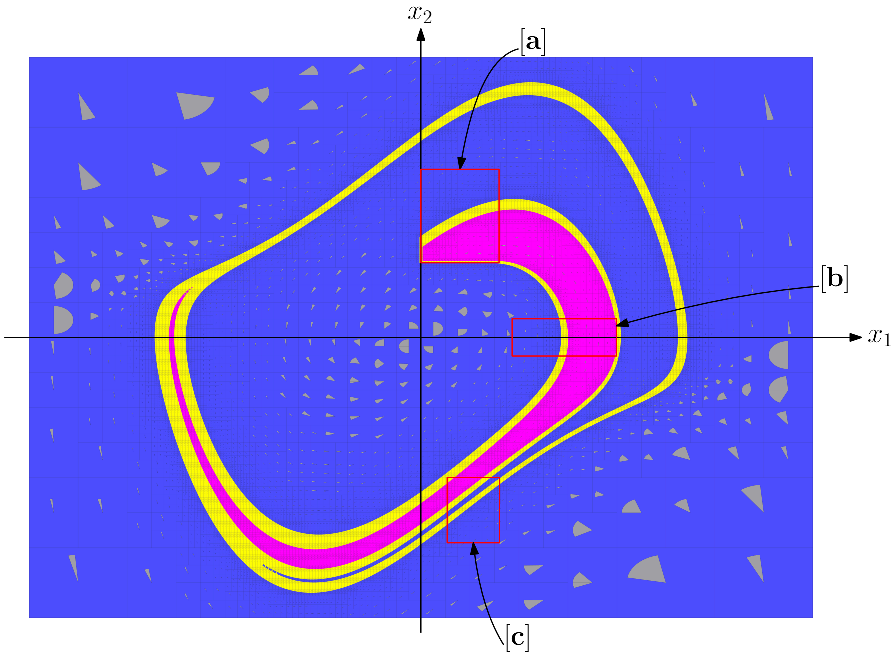

Figure 13: Feasible states associated to the Eulerian state estimation problem

In *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 753–758, December 2014.

[5] N. Delanoue, L. Jaulin, and B. Cottenceau. Attraction domain of a nonlinear system using interval analysis. In *Twelfth International Conference on Principles and Practice of Constraint Programming (IntCP 2006)*, France, Nantes, 2006.

[6] A. Desilles, H. Zidani, and E. Cruck. Collision analysis for an UAV. In *AIAA Guidance, Navigation, and Control Conference*, pages 13–16. American Institute of Aeronautics and Astronautics, 2012.

[7] L. Jaulin. *Automation for Robotics*. ISTE editions, 2015.

[8] L. Jaulin. *Mobile Robotics*. ISTE editions, 2015.

[9] L. Jaulin, D. Lopez, Le Doze, S. Le Menec, J. Ninin, G. Chabert, S. Ibn Seddik, and A. Stancu. Computing capture tubes. In M. Nehmeier, J.W. von Gudenberg, and W. Tucker, editors, *Scientific Computing, Computer Arithmetic, and Validated Numerics*, volume 9553, pages 209–224. Springer, 2016.

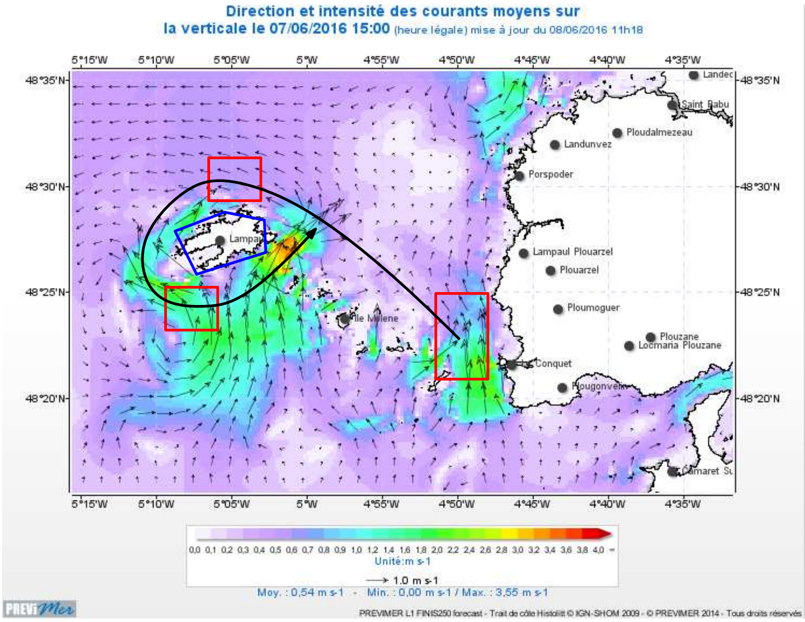

Figure 14: Visiting the three red boxes using a buoy that follows the currents is an Eulerian state estimation problem

[10] M. Konecny, W. Taha, J. Duracz, A. Duracz, and A. Ames. Enclosing the behavior of a hybrid system up to and beyond a zeno point. In *Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2013.

[11] L.D. Landau and E.M. Lifshitz. *Fluid Mechanics, Course of Theoretical Physics*. Butterworth-Heinemann, 1987.

[12] M. Lhommeau, L. Jaulin, and L. Hardouin. Capture Basin Approximation using Interval Analysis. *International Journal of Adaptative Control and Signal Processing*, 25(3):264–272, 2011.

[13] T. Le Mézo, L. Jaulin, and B. Zerr. Inner approximation of a capture basin of a dynamical system. In *Abstracts of the 9th Summer Workshop on Interval Methods*. Lyon, France, June 19-22, 2016.

[14] T. Le Mézo, L. Jaulin, and B. Zerr. An interval approach to compute invariant sets. *IEEE Transaction on Automatic Control*, 2017.

[15] Ian Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. Validating a Hamilton-Jacobi Approximation to Hybrid System Reachable Sets. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in Lecture Notes in Computer Science, pages 418–432. Springer Berlin Heidelberg, 2001.

[16] I.M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control*, pages 428–443. Springer-Verlag, 2007.

[17] D. Monnet, J. Ninin, and L. Jaulin. Computing an inner and an outer approximation of the viability kernel. *Reliable Computing*, 2016.

[18] T. Murata. Petri nets : properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[19] M. Quincampoix. Differential inclusions and target problems. *SIAM journal on control and optimization*, 30(2):324–335, 1992.

[20] N. Ramdani and N. Nedialkov. Computing Reachable Sets for Uncertain Nonlinear Hybrid Systems using Interval Constraint Propagation Techniques. *Nonlinear Analysis: Hybrid Systems*, 5(2):149–162, 2011.

[21] S. Ratschan and Z. She. Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-like Functions . *SIAM J. Control and Optimization*, 48(7):4377–4394, 2010.

[22] P. Saint-Pierre. Hybrid kernels and capture basins for impulse constrained systems. In C.J. Tomlin and M.R. Greenstreet, editors, *in Hybrid Systems: Computation and Control*, volume 2289, pages 378–392. Springer-Verlag, 2002.

[23] Patrick Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29(2):187–209, March 1994.

[24] J. Alexandre Dit Sandretto and A. Chapoutot. Validated simulation of differential algebraic equations with runge-kutta methods. *Reliable Computing*, 22, 2016.

[25] N. Seube, R. Moitie, and G. Leitmann. Aircraft Take-Off in Windshear: A Viability Approach. *Set-Valued Analysis*, 8(1-2):163–180, June 2000.

[26] Zhikun She and Bai Xue. Computing an invariance kernel with target by computing lyapunov-like functions. *IET Control Theory Applications*, 7(15):1932–1940, October 2013.

[27] W. Taha and A. Duracz. Acumen: An open-source testbed for cyber-physical systems research. In *CYCLONE'15*, 2015.

[28] D. Wilczak and P. Zgliczynski. Cr-lohner algorithm. *Schedae Informaticae*, 20:9–46, 2011.