

# Brunovsky decomposition for dynamic interval localization

Simon Rohou<sup>1</sup>, Luc Jaulin<sup>1</sup>

<sup>1</sup> ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France

**Abstract**—This paper proposes a new set-membership method for estimating the trajectories of dynamical systems, when the states are completely unknown and only non-linear observations are available. The first part of the proposed method is symbolic and follows the decomposition of Brunovsky, *i.e.*, it decomposes the set of differential equations describing the dynamical system into two blocks of constraints: one block gathers non-linear analytical equations that do not involve differential operators, while the other block is composed of linear chains of integrators. The second part of the method, that relies on the symbolic decomposition, is numerical and based on a contractor approach. It involves a specific optimal operator for narrowing the sets of feasible solutions. This approach is shown to be efficient on a difficult problem of dynamic localization of a mobile robot, without any prior knowledge about its states.

**Index Terms**—Brunovsky, non-linear system, interval analysis, localization, constraint programming, differential equations.

## I. INTRODUCTION

This paper introduces a new set-membership method for estimating the set of feasible trajectories of a dynamical system, without any prior knowledge about the states  $\mathbf{x}(t) \in \mathbb{R}^n$ , but considering a discrete set of non-linear state observations  $y(t_i) \in \mathbb{R}$ , such that:

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

$$y(t_i) = g(\mathbf{x}(t_i)) \quad (1b)$$

where  $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is a non-linear function depicting the evolution based on an input vector  $\mathbf{u}(t) \in \mathbb{R}^m$ . The observation function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$ , possibly non-linear, is assumed to be scalar without loss of generality as the method is readily scalable to the vector case. The  $t_i$ ,  $i \in \mathbb{N}$ , are observation times and the  $y(t_i)$  are the related outputs.

When the states are completely unknown, conventional methods such as Kalman filters run into difficulties, as it is difficult to find a linearization point, or to perform prediction steps. Particle filters will also miss a starting point, or will employ algorithms with high complexity without ensuring a reliable convergence. In contrast, the use of set-membership approaches avoids the need for linearization and is more suited to large uncertainties by not removing consistent solutions. They are said *reliable*, as they will always enclose feasible solutions inside guaranteed sets. In these approaches, all errors are bounded [1], [2] and an arithmetic such as interval analysis [3] allows to tackle rigorously the non-linearities of the system [4], [5].

In this context, several tools exist for estimating the set of feasible trajectories of System (1) in a reliable way. Firstly, guaranteed integration methods [6], [7], [8], [9] can be used for dealing with the non-linear differential Eq. (1a). This amounts to solving so-called Initial Value Problems (IVPs), considering some prior state vector  $\mathbf{x}_0$ , possibly enclosed in a small box  $[\mathbf{x}_0]$ . The outputs of these algorithms are guaranteed sets of feasible states. For many realistic applications, the tools available in the state-of-the-art are comfortable with the encountered non-linear equations. They will however badly behave in case of unknown initial condition, *i.e.*  $[\mathbf{x}_0]$  possibly large or unbounded, and thus are not suited for the issue addressed in this paper. Secondly, the state observations  $y(t_i)$  can be processed together with their uncertainties by means of constraint propagation tools such as contractors, that are operators for safely narrowing the bounds of the state space according to a given constraint (*e.g.* an equation), without removing feasible solutions from the sets. However, if one considers System (1) with unbounded prior sets for enclosing the  $\mathbf{x}(t)$  together with only few non-linear observations, the use of state-of-the-art contractor methods leads to large sets that are not relevant for state estimation purposes. In order to overcome this problem, some branch-and-prune techniques such as shaving methods [10] can be applied in order to reduce the state sets by performing bisections. The counterpart is obviously the increasing complexity of these algorithms. As a consequence, the current existing tools are not sufficient for addressing the considered problem both in a reliable and an efficient way.

Our contribution relies on the decomposition of the non-linear dynamical system into a Brunovsky canonical form [11], [12]. The obtained decomposition is possible for flat systems which constitutes a huge class of vehicles [13] such as cars [14], mobiles with trailers [15], sailboats [16], *etc.* We will show that the obtained Brunovsky form makes it possible to solve our localization problem efficiently and in a reliable way. Besides, as this method does not involve classical interval integration tools [17], [18], [19], [20], [21], [22], the contribution of this paper brings a new approach for dealing with guaranteed integration.

The paper is organized as follows. Sections II and III recall the principles of flat systems and Brunovsky decomposition. Section IV outlines set-membership methods involving contractors. It introduces the notion of Contractor Circuit which allows a better understanding of the contraction process.

Section V details a new contractor for dealing with chains of integrators, resulting from the Brunovsky decomposition. Finally, Section VI shows how to apply the proposed method for solving a state estimation problem such as the localization of a mobile robot. Convincing results are provided in this section on a reproducible robotic example. Section VII concludes the paper.

## II. FLAT SYSTEMS

Consider Eq. (1a) involved in the following system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{z} = \mathbf{h}(\mathbf{x}), \end{cases} \quad (2)$$

where  $\mathbf{z} \in \mathbb{R}^m$  is the output vector used with a control point of view, *i.e.*,  $\mathbf{z}$  is not necessarily measured. Both  $\mathbf{f}$  and  $\mathbf{h}$  are assumed to be smooth. The dimension of  $\mathbf{u}$  and  $\mathbf{z}$  are assumed to be both equal to  $m$ . The system is said to be flat with the flat output  $\mathbf{z}$  [23], [24], if there exists two continuous functions  $\phi$  and  $\psi$  and integers  $\kappa_1, \dots, \kappa_m$  such that

$$\begin{cases} \mathbf{x} = \phi \left( z_1, \dot{z}_1, \dots, z_1^{(\kappa_1-1)}, \dots, z_m, \dot{z}_m, \dots, z_m^{(\kappa_m-1)} \right) \\ \mathbf{u} = \psi \left( z_1, \dot{z}_1, \dots, z_1^{(\kappa_1)}, \dots, z_m, \dot{z}_m, \dots, z_m^{(\kappa_m)} \right). \end{cases} \quad (3)$$

The integers  $\kappa_i$ , called the Kronecker indexes, are unique and should be such that  $\kappa_1 + \dots + \kappa_m = n$ , [25]. A system satisfying Eq. (3) is observable: the function  $\phi$  gives us the unique state vector consistent with the flat outputs and their derivatives. In practice, the functions  $\phi$  and  $\psi$  involved in Eq. (3) can be obtained by following these steps:

- The derivation step (see [26]) computes symbolically  $z_1, \dot{z}_1, \dots, z_1^{(\kappa_1)}, \dots, z_m, \dot{z}_m, \dots, z_m^{(\kappa_m)}$  as functions of  $\mathbf{x}$  and  $\mathbf{u}$ , using Eq. (2). We obtain an expression of the form

$$\begin{pmatrix} z_1 \\ \dot{z}_1 \\ \vdots \\ z_m^{(\kappa_m)} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}. \quad (4)$$

This can be done automatically without any difficulty using Lie derivatives. It suffices to take all  $m$  equations  $z_j = h_j(\mathbf{x})$  and to compute symbolically its first, second,  $\dots$   $\kappa_j$ th derivatives with respect to  $t$ . At each step, the  $\dot{x}_i$  are replaced by  $f_i(\mathbf{x}, \mathbf{u})$ .

- The inversion step has to be performed to obtain an expression of the form of Eq. (3) from the inversion of  $\lambda$ . This operation is difficult to obtain except for simple systems. The inversion step will not be required in our approach as it will be implicitly processed by our interval-based resolution.

**Example 1.** Consider the system

$$\begin{cases} \dot{x}_1 = x_1 + x_2 \\ \dot{x}_2 = x_2^2 + u \\ z = x_1. \end{cases} \quad (5)$$

For the derivation step, we compute  $z, \dot{z}, \ddot{z}, \dots$  with respect to  $\mathbf{x}$  and  $u$  until  $u$  occurs. We get

$$\begin{cases} z = x_1 \\ \dot{z} = \dot{x}_1 = x_1 + x_2 \\ \ddot{z} = \dot{x}_1 + \dot{x}_2 = x_1 + x_2 + x_2^2 + u. \end{cases} \quad (6)$$

Since we had to derive twice, we conclude that the Kronecker index is  $\kappa = 2$  which corresponds to the dimension of  $\mathbf{x} = (x_1, x_2)^\top$ . As a consequence, the output  $z$  is flat. From (6) we read

$$\lambda \begin{pmatrix} \mathbf{x} \\ u \end{pmatrix} = \begin{pmatrix} x_1 \\ x_1 + x_2 \\ x_1 + x_2 + x_2^2 + u \end{pmatrix}. \quad (7)$$

For the inversion step, we have to isolate  $\mathbf{x}$  and  $u$  in order to obtain an expression with respect to  $z, \dot{z}, \ddot{z}$ . The inversion is easy for this example:

$$\begin{cases} x_1 = z \\ x_2 = \dot{z} - x_1 = \dot{z} - z \\ u = \ddot{z} - (x_1 + x_2 + x_2^2) = \ddot{z} - \dot{z} - (\dot{z} - z)^2. \end{cases} \quad (8)$$

As a consequence,

$$\begin{cases} \phi(z, \dot{z}) = \begin{pmatrix} z \\ \dot{z} - z \end{pmatrix} \\ \psi(z, \dot{z}, \ddot{z}) = \ddot{z} - \dot{z} - (\dot{z} - z)^2. \end{cases} \quad (9)$$

We can therefore consider that we have a procedure to find the functions  $\phi$  and  $\psi$  as soon as we have a flat output  $\mathbf{z}$ . Unfortunately, to our knowledge, there is no general method to find a flat output for a given non-linear system [27]. In practice, we can use the physical intuition we have on the system [15] or tools coming from differential algebra such as the approach based on finitely generated differential ideal [28].

## III. BRUNOVSKY DECOMPOSITION

The differential flat system  $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$  with flat outputs  $z_1, \dots, z_m$  and sensor outputs  $\mathbf{y}$  admits the following Brunovsky decomposition:

$$\begin{cases} \begin{pmatrix} z_1 \\ \dot{z}_1 \\ \vdots \\ z_m^{(\kappa_m)} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix} \\ z_1^{(\kappa_1)} \xrightarrow{f_1} \dots \xrightarrow{f_1} \dot{z}_1 \xrightarrow{f_1} z_1 \\ \vdots \\ z_m^{(\kappa_m)} \xrightarrow{f_m} \dots \xrightarrow{f_m} \dot{z}_m \xrightarrow{f_m} z_m \\ \mathbf{y} = \mathbf{g}(\mathbf{x}) \end{cases} \quad (10)$$

This decomposition clearly makes appear the Brunovsky normal form through the chain of integrators. It is illustrated by Fig. (1).

**Example 2.** Let us consider the system described by the following equations:

$$\begin{cases} (i) \quad \dot{x}_1 = x_4 \cos(x_3) \\ (ii) \quad \dot{x}_2 = x_4 \sin(x_3) \\ (iii) \quad \dot{x}_3 = u_1 \\ (iv) \quad \dot{x}_4 = u_2 \end{cases} \quad (11)$$

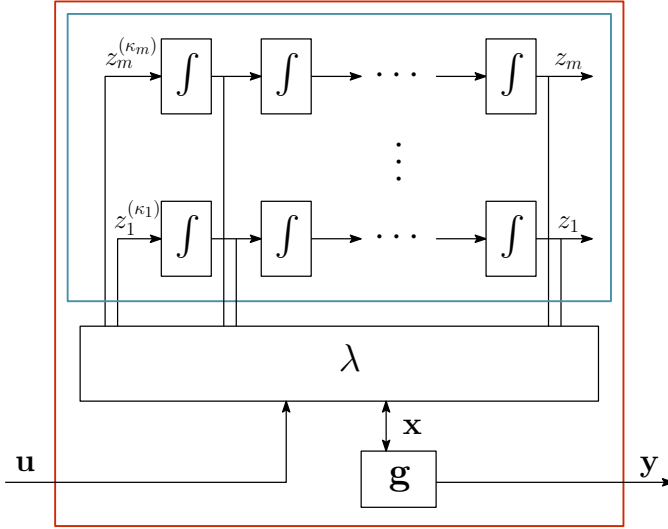


Fig. 1. Non-linear dynamical Systems (1) and (2) transformed into a Brunovsky canonical form. All the differential relations are gathered in chains of integrators. The rest of the system is expressed by means of non-differential functions  $\lambda$  and  $g$ .

This system corresponds to a wheeled robot moving on a plane.  $(x_1, x_2)^\top$  is its location,  $x_3$  its heading,  $x_4$  its speed and  $\mathbf{u} \in \mathbb{R}^2$  is the control vector: acceleration and rotation speed. This system is known to be flat [15], if we take as flat output  $\mathbf{z} = (x_1, x_2)^\top$ . To check this, we compute the Brunovsky form, applying the method of feedback linearization as explained in [26]. We get that  $\kappa_1 = \kappa_2 = 2$ , and thus we have to differentiate  $\mathbf{z}$  twice. From Eq. (11)–(i) and (11)–(ii), we obtain the following Brunovsky decomposition in which static (*i.e.* non-differential) constraints (I) are separated from chains of integrators (II):

$$(I) \quad \begin{cases} \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_4 \cos(x_3) \\ x_4 \sin(x_3) \\ u_2 \cos(x_3) - u_1 x_4 \sin(x_3) \\ u_2 \sin(x_3) + u_1 x_4 \cos(x_3) \end{pmatrix} \\ \lambda(\mathbf{x}, \mathbf{u}) \end{cases}$$

$$(II) \quad \begin{cases} \ddot{z}_1 \xrightarrow{f} \dot{z}_1 \xrightarrow{f} z_1 \\ \ddot{z}_2 \xrightarrow{f} \dot{z}_2 \xrightarrow{f} z_2 \end{cases} \quad (12)$$

Although Block (I) in (12) appears as a differential equation, the expressions of each of the six variables  $z_1, z_2, \dots, \ddot{z}_2$  are considered independently and under the form of pure algebraic constraints, so that all the differential part is delegated to Block (II) in which these variables are link to each other. As expected, we have  $\kappa_1 + \kappa_2 = 4$ , which is consistent with the dimension of the system. Note that the Brunovsky Form (12) and the state Eq.s (11) are equivalent.

This section has provided an illustration of how differential equations describing a dynamical system can be separated into

two blocks of constraints. The first block (I) corresponds to non-linear equations with no differential operator inside, while the second block (II) is only made of chains of integrators. The following section shows how these blocks can be dealt with using contractor techniques.

#### IV. SET-MEMBERSHIP METHOD

The previous decomposition provides a formulation for a dynamical system which will be shown to be adapted to set-membership techniques. The relations from the decomposition are sometimes called *constraints*. In a set-membership context, we can formalize a state estimation problem as a Constraint Network (CN), see for instance [29], [30]. Interval operators, called *contractors*, are used to contract the set of feasible values of the state variables of the system.

##### A. Contractors

Contractors are algorithms based on interval analysis, that allow reliable computations despite non-linearities, large uncertainties or singularities. This arithmetic ensures to never remove valid solutions from the domains of the variables, which is critical for the safety of complex systems.

When using the framework of contractors, state estimation consists in contracting sets defined beforehand. Several contractors can be called on a set until a fixed point is reached, *i.e.* when the set cannot be reduced anymore. In practice, a set represents the knowledge we have about a vector or a trajectory. When nothing is known beforehand, as it is the case for the state vectors  $\mathbf{x}$  in the considered problem, then unbounded sets are defined, such as infinite intervals or boxes  $[\mathbf{x}] = [-\infty, \infty]^n$ . The same approach applies for trajectories by using *tubes*, *i.e.* intervals of trajectories denoted by  $[\mathbf{x}](\cdot)$ , which allows to process temporal or differential equations elegantly by processing one single object. Fig. 2 illustrates a scalar tube enclosing a feasible trajectory.

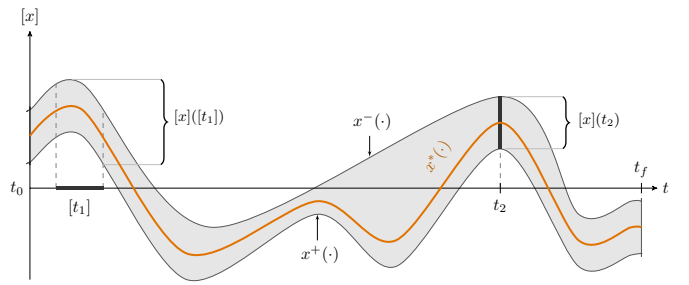


Fig. 2. A tube  $[x](\cdot)$  defined on the temporal domain  $[t_0, t_f]$  as an interval of two signals  $[x^-(\cdot), x^+(\cdot)]$  and enclosing an unknown trajectory  $x^*(\cdot)$ . The thinner the tube, the better the approximation of  $x^*(\cdot)$ .

A collection of contractors exists in the literature (see for instance [31], [32], [33]) and can be found in open-source libraries such as Codac [34]. Each of these tackles a given equation and provides an algorithm ready to be applied on sets. When a set is provided as input of such algorithm, it is possibly contracted and thus some infeasible values are removed. For instance, the equation  $\mathbf{f}(\mathbf{x}) = \mathbf{0}$  can be addressed by a contractor  $C_{\text{HC4revise}}$  provided in [35] and applied on some

box  $[\mathbf{x}]$ . The obtained contracted box  $\mathcal{C}_{\text{HC4revise}}([\mathbf{x}])$  is a subset of  $[\mathbf{x}]$  but still contains all feasible vectors consistent with  $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ . This consistency property of contractors allows to apply them as much as is necessary. Several contractors can be combined in order to tackle more complicated problems. From Tarski's theorem [36] and since all contractors we use are inclusion monotonic, the resulting propagation method always converges.

### B. Contractor Circuits

Successive contractions allow to converge towards accurate outer approximations of sets of feasible solutions for the considered problem. This is achievable by combining several contractors on common sets. Usually, a decomposition of the problem is performed in order to bring out equations associated with already available contractors. For instance, the Brunovsky Form (10) already provides a decomposition for which contractors are at hand. Graphically, the contractor resolution amounts to the *Contractor Circuit* depicted in Fig. 3.

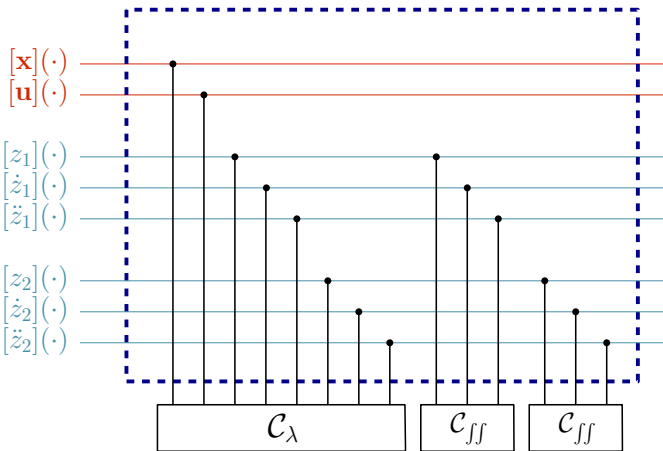


Fig. 3. So-called *Brunovsky Circuit* associated with the Brunovsky decomposition of the robotic Eq. (12).

The concept of *Contractor Circuit*, introduced in this paper, allows a better understanding of the use of contractors for state estimation. A Contractor Circuit is a Boolean matrix that links contractors to sets. In this representation, entries of the matrix are represented by the small black dots at the intersection of horizontal and vertical lines. At the bottom of the figure, each rectangular block corresponds to a contractor involved in the circuit. Horizontal lines are associated with sets, *i.e.* tubes in this example. The red horizontal lines correspond to the inputs and the blue ones to the output variables. The black vertical lines represent the possible inputs of the contractor algorithms.

The initial bounds of the tubes correspond to some prior knowledge we have on the values, or to some measurements that have been performed. The latter can be seen as a *restriction* on the sets: the measurement is directly encoded in the set. Tubes are therefore initialized to  $[-\infty, \infty], \forall t$  if there is not prior information available for the related variables, as it is the case for the state trajectory  $\mathbf{x}(\cdot)$ . The contractors involved in the circuit are then called until no more contraction can

be observed. Note that the order in which the contractors are called will not affect the final results [37], because all the operators are called indefinitely until a fixed point is reached. The use of a Boolean matrix can be avoided but the call of the contractors and the connections between tubes should then be done by hand inside the program. Using the connection matrix allows us to enclose a part of the usual code inside the data structure of the problem. As a consequence, it helps to define a high level language for computing with contractors.

The Circuit depicted in Fig. 3 involves the contractor  $\mathcal{C}_\lambda$  related to the non-linear equation  $(\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}})^\top = \lambda(\mathbf{x}, \mathbf{u})$ . The literature already provides algorithms for addressing this relation, see for instance the `HC4revise` contractor introduced in [35]. In addition, the circuit involves two  $\mathcal{C}_{ff}$  operators that are the contractors for dealing with integrator chains. Because the dimension of  $\mathbf{z}$  is equal to 2, two contractor chains are involved in this circuit.  $\mathcal{C}_{ff}$  cannot be directly found in the literature, but is required for Brunovsky decompositions. The following section proposes an algorithm for  $\mathcal{C}_{ff}$ .

### V. THE INTEGRATOR CHAIN CONTRACTOR

In order to process the Brunovsky decomposition in a set-membership context, a dedicated integrator chain contractor, denoted by  $\mathcal{C}_{ff}$ , has to be provided. As explained in Section III, the integrator chain encompasses all differential links of System (1) in one single relation given by:

$$z^{(\kappa)} \xrightarrow{f} \dots \xrightarrow{f} \dot{z} \xrightarrow{f} z \quad (13)$$

where  $z$  is a scalar output as defined in System (2). For vector outputs  $\mathbf{z}$ , Relation (13) is involved several times according to the dimension of  $\mathbf{z}$ . This integrator chain allows to accurately propagate information from one signal through its primitives and derivatives. Note that this relation could be broken down into several elementary integrators as in (14), for which contractors already exist, see for instance [38].

$$z^{(\kappa)} \xrightarrow{f} z^{(\kappa-1)}, \dots, \ddot{z} \xrightarrow{f} \dot{z}, \dots, \dot{z} \xrightarrow{f} z. \quad (14)$$

However, in a set-membership context, this leads to well-known *wrapping effects* due to the enclosure of each intermediate result into non-optimal sets. The use of one single operator allows optimal computations.

#### A. Linear state estimator

The chain (13) can be expressed under the form of a specific linear state estimation for which the state is defined by a set  $\mathbf{z}$  of primitives/derivatives of one only signal<sup>1</sup> and for which the observations are provided by restrictions on these primitives/derivatives. More precisely, the integrator chain constraint involving the signals  $(z^{(0)}, z^{(1)}, \dots, z^{(\kappa)}, w)$  and defined as:

$$w \xrightarrow{f} z^{(\kappa)} \xrightarrow{f} \dots \xrightarrow{f} \dot{z} \xrightarrow{f} z \quad (15)$$

<sup>1</sup>The notation  $\mathbf{z}$  is chosen voluntarily as it will refer, in the application of the contractor related to this state estimator, to the output of System (2).

can be cast into the following linear system corresponding to the Brunovsky form:

$$\dot{\mathbf{z}}(t) = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \\ \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}}_{\mathbf{A}} \mathbf{z}(t) + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}}_{\mathbf{B}} w(t), \quad (16)$$

where  $w(\cdot)$  is known to be inside a tube  $[w](\cdot)$ . An integrator chain contractor  $\mathcal{C}_{ff}([z](\cdot), [w](\cdot))$  has to be built to contract a tube  $[z](\cdot) = ([z^{(0)}](\cdot), [z^{(1)}](\cdot), \dots, [z^{(\kappa)}](\cdot))^T$  from Eq. (16) and  $[w](\cdot)$ .

In the set-theoretical framework, due to the linearity of System (16), an optimal contraction is achievable by computing a set of abstract sets  $\mathbb{Z}(t)$  over time, using the following equations:

$$\begin{aligned} \vec{\mathbb{Z}}(t) &\subset \check{\mathbb{Z}}(t) \cap \left( e^{\mathbf{A}\delta} \cdot \vec{\mathbb{Z}}(t-\delta) + \delta e^{\mathbf{A}\cdot[0,\delta]} \cdot \mathbf{B} \cdot [w]([t-\delta, t]) \right) \\ \overleftarrow{\mathbb{Z}}(t) &\subset \check{\mathbb{Z}}(t) \cap \left( e^{-\mathbf{A}\delta} \cdot \overleftarrow{\mathbb{Z}}(t+\delta) - \delta e^{-\mathbf{A}\cdot[0,\delta]} \cdot \mathbf{B} \cdot [w]([t, t+\delta]) \right) \\ \mathbb{Z}(t) &= \vec{\mathbb{Z}}(t) \cap \overleftarrow{\mathbb{Z}}(t) \end{aligned}$$

where

- $\check{\mathbb{Z}}(t)$  is the prior set of states for  $t$ ;
- $\vec{\mathbb{Z}}(t)$ ,  $\overleftarrow{\mathbb{Z}}(t)$  are the sets of all states at  $t$  consistent with the past and the future, respectively;
- $\mathbb{Z}(t)$  is the smallest set for  $\mathbf{z}(t)$  assuming that  $w(t) \in [w](t)$  and  $\mathbf{z}(t) \in \check{\mathbb{Z}}(t)$ . It is the result of the contraction.

It has been shown that this observer is exact in the discrete time case, if we represent the sets  $\check{\mathbb{Z}}(t)$ ,  $\vec{\mathbb{Z}}(t)$ ,  $\overleftarrow{\mathbb{Z}}(t)$ ,  $\mathbb{Z}(t)$  by polygons [39], [40]. An implementation in the continuous-time case has been recently proposed in [41]. The principle is to use a tube for representing prior sets  $\check{\mathbb{Z}}(\cdot)$  and to apply the exact discrete time observer of [39] for each slice of the tube. The resulting contractor is still minimal if we assume that we can neglect the discretization errors, otherwise, an unavoidable small pessimism exists, due to the implementation of tubes.

Therefore, state observations are processed as restrictions on the  $\check{\mathbb{Z}}(t)$  prior sets. In our approach, these restrictions directly come from the tubes, under the form of boxes, since our tubes are implemented as sets of boxed slices. The contraction results into the intersection between the integration of the prior sets  $\check{\mathbb{Z}}(t-\delta)$  into  $\vec{\mathbb{Z}}(t)$  (from Eq. (16)) and the prior sets  $\check{\mathbb{Z}}(t)$ . It always corresponds to the intersection of a polygon and a box, which can be computed accurately. Fig. 4 provides a simple illustration of this state estimation in the continuous case, using polygons and tubes. Considering for instance the chain  $w \xrightarrow{f} z_2 \xrightarrow{f} z_1$ , prior 2d sets  $\check{\mathbb{Z}}(\cdot)$  are implemented as tubes  $[z_1] \times [z_2](\cdot)$  (upper part of the figure). The lower part of the figure depicts in gray one integration step (polygon  $\check{\mathbb{Z}}(t_1)$ ) into polygon  $\vec{\mathbb{Z}}(t_1+\delta)$  and the intersection with the prior set at time  $t_1+\delta$ : the box  $\check{\mathbb{Z}}(t_1+\delta)$ . The resulting contracted set, represented in hatched blue, is also a polygon.

The guaranteed linear integration transforms each vector into a box and each box or polygon into a polygon. The use of polygons for representing sets allows to obtain optimal results

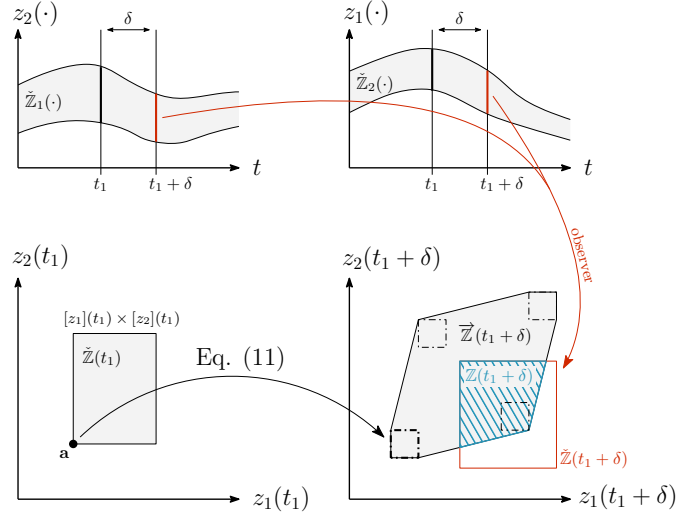


Fig. 4. Example of integrator chain contractor over one time step.

with minimal enclosures. However, it must be highlighted that this optimality comes with a strong complexity of the algorithms, since the number of vertices increases dramatically with time. Non-minimal enclosures could be used instead, such as intervals [42], [43], zonotopes [44] or parallelotopes [45], allowing faster computations. In practice, these enclosures would be more suited for state estimation, since the optimality of this differential contractor is not always relevant with respect to the number of observations of the system. A first study has recently led to the publication of [46] in which an ellipsoidal contractor is provided for  $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{w}$  with higher performances (faster computations, implementation for any dimension) than the polygonal approach of [41], while maintaining a quite accurate envelope of the states. Its application in the context of this paper remains to be studied, but is encouraging for real time applications.

As a conclusion, a contractor dealing with linear systems expressed as  $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{w}$  can be configured in order to deal with a chain of integrators. This allows to contract a list of tubes directly linked by their derivative and primitive relations. In the remaining of this paper, the contractor provided in the publication [41] is used for the  $\mathcal{C}_{ff}$  involved in the Brunovsky circuit.

## VI. APPLICATION TO LOCALIZATION

The set-membership Brunovsky decomposition proposed in this paper is now illustrated on a difficult academic robotic problem.

### A. Problem statement

We consider the localization problem of a robot moving in an environment where few landmarks are available [47], and formalized by System (1). This state estimation is difficult if all state vectors  $\mathbf{x}(t) \in \mathbb{R}^4$ , describing the positions  $(x_1, x_2)^T$ , headings  $x_3$  and speeds  $x_4$  of the vehicle are unknown, *i.e.* when initial conditions or direct measurements of the states are not at hand. Only the control  $\mathbf{u}(t) \in \mathbb{R}^2$  of the system

(linear acceleration and rotation rate) is known, with some bounded uncertainties, together with some interval range-only observations  $[y](t_i)$  made between the moving vehicle and two landmarks, for which the positions  $\mathbf{m}^a$  and  $\mathbf{m}^b$  are known.

The goal of this problem is to perform a full state estimation by computing an approximation of the four components of  $\mathbf{x}(t)$  along time, including the headings  $x_3(t)$ . Such state estimation is difficult due to the absence of knowledge on the initial condition  $\mathbf{x}(0)$  as well as any other state, and because very few asynchronous non-linear observations are available. In particular, it has been shown in [48] and [49], that the problem is difficult to solve if the heading of the robot is not measured. As it has been said before, classical methods such as Bayesian approaches or particle filters badly behave in presence of few observations and are not suited to deal with large uncertainties because of inaccurate linearizations or convergence issues. On the other hand, set-membership approaches are more dedicated to limited observations and non-linear contexts, but very few studies have been done on this application [29]. One should note the work of [50] based on interval methods with a pruning component [51], that we want to avoid. Indeed in [50], the approximation of heading values is obtained by bisecting the heading space. Therefore, a large number of bisections is needed in order to obtain a significant approximation [52], which ruins the performances of interval methods [53], [54].

### B. Reproducible test case

Let us consider the following reproducible example. The proposed simulation is based on analytical expressions and simple data in order to encourage future comparisons with the method provided in this paper. A wheeled robot described by Eq. (11) follows a planar trajectory given by a Lissajous equation. The unknown states are given by:

$$\mathbf{x}(t) = \begin{pmatrix} 10 \cos(t) \\ 5 \sin(2t) \\ \text{atan2}(10 \cos(2t), -10 \sin(t)) \\ \sqrt{(-10 \sin(t))^2 + (10 \cos(2t))^2} \end{pmatrix}. \quad (17)$$

Eq. (17) cannot be used during the resolution. It is only given to validate the state estimation, and also to generate data (exteroceptive measurements and inputs). From Eq. (17), we obtain the following inputs:

$$\mathbf{u}(t) = \begin{pmatrix} \frac{2 \sin(t) \sin(2t) + \cos(t) \cos(2t)}{\sin^2(t) + \cos^2(2t)} \\ \frac{10 \cos(t) \cdot \sin(t) - 20 \cos(2t) \cdot \sin(2t)}{\sqrt{\sin^2(t) + \cos^2(2t)}} \end{pmatrix}. \quad (18)$$

The inputs are supposed accurately known. In our method, they are actually numerically enclosed in a thin tube  $[\mathbf{u}](\cdot)$  that is not degenerated because of the sliced implementation of the tubes of the used library [34].

Finally, we assume that the robot performs distance measurements between its position and one of the landmarks  $\mathbf{m}^a$  or  $\mathbf{m}^b$ , respectively located at  $(-5, 6)$  and  $(0, -4)$ . The observation equation (1b) is therefore the distance function:

$$y^j(t_i) = \sqrt{(x_1(t_i) - m_1^j)^2 + (x_2(t_i) - m_2^j)^2}. \quad (19)$$

The beacons emit asynchronously range-only signals received by the robot. These bounded measurements are given in Table I.

TABLE I  
SET OF FOUR BOUNDED MEASUREMENTS  $(t_i, [y](t_i))$ .

$t_i$	$[y^a](t_i)$	$t_i$	$[y^b](t_i)$
0.75	[12.333, 12.383]	1.50	[4.733, 4.783]
2.25	[10.938, 10.988]	3.00	[10.211, 10.261]

The simulation is run for  $t \in [0, 3]$ .

### C. Interval Brunovsky localization: methodology

Our interval state estimation approach consists in the following steps:

- First, a symbolic decomposition of Eq. (11) is made in order to obtain a Brunovsky form. The decomposition makes appear two blocks of differential and non-differential relations, as in (12).
- Second, sets of feasible values are defined in order to enclose the involved variables  $\mathbf{x}(\cdot)$ ,  $\mathbf{u}(\cdot)$ , and the  $[y](t_i)$ 's. For the states, we define an unbounded tube  $[\mathbf{x}](\cdot)$  such that  $\forall t, [\mathbf{x}](t) = [-\infty, \infty]^4$ . The tube  $[\mathbf{u}](\cdot)$  is thin and centered on Eq. (18). This means that the analytical knowledge of  $\mathbf{u}(\cdot)$  is not required by our method<sup>2</sup>. Finally, interval measurements  $[y](t_i)$  are given by Table I.
- The third step consists in defining contractors for the equations resulting from the first step. This amounts to build the *Brunovsky Circuit* of Fig. 3. The non-differential and non-linear function  $\lambda(\mathbf{x}, \mathbf{u})$  is given by (12) and a contractor  $\mathcal{C}_\lambda$  can be easily built from this analytical expression [35]. Then, the two  $\mathcal{C}_{ff}$  of Fig. 3 are contractors for dealing with linear observers, such as those provided in [41] or [46], and configured according to System (16). We add another contractor  $\mathcal{C}_g$  for dealing with Eq. (19), also built from an analytical expression. This last contractor links some slices  $[\mathbf{x}_{1,2}](t_i)$  of the tube  $[\mathbf{x}](\cdot)$  with the measurements  $[y](t_i)$ . More details are given in [32].
- Finally, the contractors are applied on the intervals and tubes defined in the second step. Some iterative resolution is performed by calling all the contractors until no more contraction is obtained. It should be recalled that the order of calls does not change the quality of the results, only the computation time may differ. Once this fixed point has been reached, the estimated (feasible) states are enclosed in the contracted tube  $[\mathbf{x}](\cdot)$ .

### D. Results

The obtained results for the position tube  $[\mathbf{x}_{1,2}](\cdot)$  is illustrated in blue in Fig. (5). It encloses the unknown true

<sup>2</sup>Because our tubes are implemented as lists of boxes, some pessimism related to wrapping effects is introduced. The time-discretization of our tubes is set to  $\delta = 0.003$ .

trajectory given by Eq. (17) that is white painted. As one can see, despite the absence of prior knowledge on the states, the obtained envelope is thin thanks to the use of efficient contractors. The two landmarks are pictured by orange boxes and the four range-only measurements are drawn by gray lines.

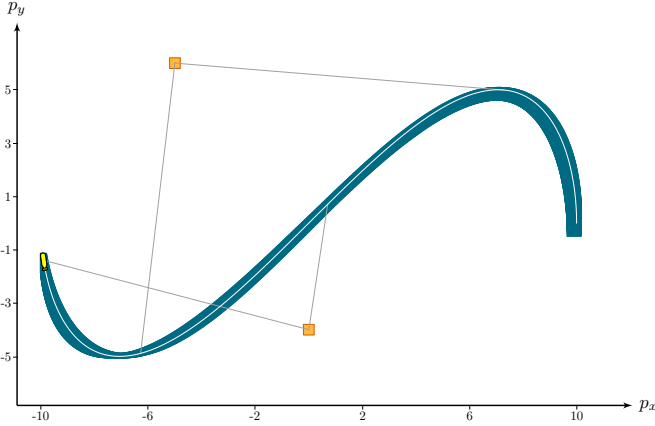


Fig. 5. Set  $[x](\cdot)$  of feasible states projected in two dimensions. The unknown planar trajectory remains enclosed in the tube, which illustrates the guarantee of the computations.

The simulation, run in 36 seconds, also provides a contraction of the velocity tube  $[x_4](\cdot)$  and the heading tube  $[x_3](\cdot)$ , as illustrated by Fig. 6. The example has been processed by operators delivered in the Codac library [34], in which this example is freely available. The Reader may find more material related to this work on <http://codac.io/brunovsky>.

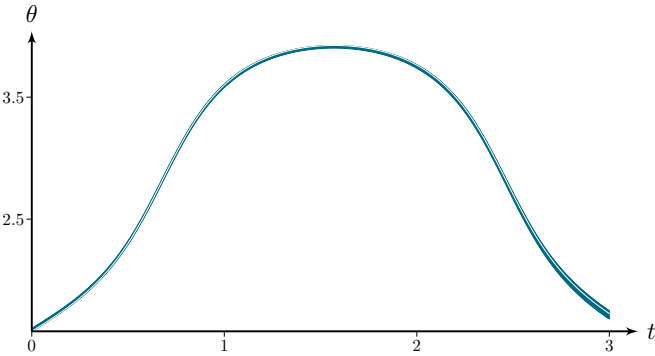


Fig. 6. The tube  $[x_3](\cdot)$  of feasible headings. The actual but unknown truth, given by  $x_3^*(t) = \text{atan2}(10 \cos(2t), -10 \sin(t))$ , is plotted in white and guaranteed to be enclosed in the computed  $[x_3](\cdot)$ .

## VII. CONCLUSIONS

This paper has presented a new approach for state estimation of non-linear dynamical systems, that combines some non-linear state estimation techniques [55] based on flatness [15] with interval contractors. Flatness allows us to build a Brunovsky decomposition made with a set of static non-differential and non-linear relations, together with chains of integrators. A contractor based approach is then able to take advantage of this decomposition in order to enclose, in a guaranteed way, the unknown states into an accurate tube.

In our set-membership approach, all the differential relations are processed by one single operator, able to deal with a chain of integrators, and providing optimal results when using polygons as wrappers. The resulting state estimator has several advantages over classical approaches:

- The state estimator *is reliable with respect to non-linearities*. Thanks to interval analysis, it is able to deal with non-linear (or non-differentiable and even non-continuous) state equations, without linearizing (as done by the extended Kalman filter [56]) or approximating them.
- The state estimator *does not require the interval integration of differential equations*. Such integrations are needed by all other interval state estimation methods [21], [8], [57], [58], which makes them inefficient for large uncertainties.
- The state estimator *can be used for real-time applications*.

Our approach has been applied to the range-only localization of a robot, without any measurements of its heading or speed, nor any knowledge about its positions. This state estimation problem, that is known to be difficult to address, has been accurately and reliably solved using the Brunovsky decomposition and the specific operators described in this paper, without any bisection technique or prior knowledge on the states.

## ACKNOWLEDGMENT

This work was supported by the French Agence Nationale de la Recherche (ANR) [grant number ANR-16-CE33-0024].

## REFERENCES

- [1] E. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*. London, UK: Springer-Verlag, 1997.
- [2] E. Walter and H. Piet-Lahanier, "Estimation of the parameter uncertainty resulting from bounded-error data," *Mathematical Biosciences*, vol. 92, pp. 55–74, 1988.
- [3] R. B. Kearfott and V. Kreinovich, Eds., *Applications of Interval Computations*. Dordrecht, the Netherlands: Kluwer, 1996.
- [4] D. Meizel, O. Lévêque, L. Jaulin, and E. Walter, "Initial localization by set inversion," *IEEE transactions on robotics and Automation*, vol. 18, no. 6, pp. 966–971, 2002.
- [5] B. Desrochers, S. Lacroix, and L. Jaulin, "Set-membership approach to the kidnapped robot problem," in *IROS 2015*, 2015.
- [6] N. Nedialkov, "VNODE-LP, A Validated Solver for Initial Value Problems in Ordinary Differential Equations," Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada, Tech. Rep. CAS-06-06-NN, 2006.
- [7] T. Kapela, M. Mrozek, P. Pilarczyk, D. Wilczak, and P. Zgliczynski, "CAPD – a rigorous toolbox for Computer Assisted Proofs in Dynamics," <http://capd.ii.uj.edu.pl/>, 2010.
- [8] N. Revol, K. Makino, and M. Berz, "Taylor models and floating-point arithmetic: proof that arithmetic operations are validated in COSY," *Journal of Logic and Algebraic Programming*, vol. 64, pp. 135–154, 2005.
- [9] A. Chapoutot, J. A. dit Sandretto, and O. Mullier, *Dynibex, available at*, <http://perso.ensta-paristech.fr/chapoutot/dynibex/>. ENSTA, 2015.
- [10] G. Trombettoni and G. Chabert, "Constructive Interval Disjunction," in *Proc. CP, Constraint Programming*, LNCS 4741, 2007, pp. 635–650.
- [11] P. Brunovsky, "A classification of linear controllable systems," *Kybernetika*, vol. 6, no. 3, pp. 173–188, 1970.
- [12] E. Sontag, *Mathematical control theory: deterministic finite dimensional systems (2nd ed.)*. Berlin: Springer-Verlag, 1998.
- [13] P. Martin and P. Rouchon, "Any (controllable) driftless system with m inputs and m + 2 states is flat," in *Proc. of Conf. on Decision and Control*, 1995.

- [14] S. Sekhavat and J. Hermosillo, "The cycab robot is a differentially flat system," *In Proc. of Intelligent Robots and Systems, IROS'00, Japan 2000*, 2000.
- [15] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and applications," *International Journal of Control*, no. 61, pp. 1327–1361, 1995.
- [16] L. Jaulin, "Combining interval analysis with flatness theory for state estimation of sailboat robots," *Mathematics in Computer Science*, vol. 6, no. 4, pp. 247–259, 2012.
- [17] W. Tucker, "A Rigorous ODE Solver and Smale's 14th Problem," *Foundations of Computational Mathematics*, vol. 2, no. 1, pp. 53–117, 2002.
- [18] E. Goubault, O. Mullier, S. Putot, and M. Kieffer, "Inner approximated reachability analysis," in *HSCC '14*, Berlin, Germany, Apr. 2014, pp. 163–172.
- [19] A. Goldsztejn and W. Hayes, "Reliable inner approximation of the solution set to initial value problems with uncertain initial value," in *SCAN, Duisburg, Germany, September 2006*, 2006.
- [20] J. A. dit Sandretto and A. Chapoutot, "Validated explicit and implicit Runge-Kutta methods," *Reliable Computing*, vol. 22, pp. 79–103, 2016.
- [21] D. Wilczak and P. Zgliczynski, "Cr-lohner algorithm," *Schedae Informaticae*, vol. 20, pp. 9–46, 2011.
- [22] N. Ramdani and N. Nedialkov, "Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint propagation techniques," *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 149–162, 2011.
- [23] S. Diop and M. Fliess, "Nonlinear observability, identifiability and persistent trajectories," in *Proc. 36th IEEE Conf. Decision Control*, Brighton, 1991, pp. 714–719.
- [24] J. Gauthier and I. Kupka, *Deterministic observation theory and applications*. Cambridge University Press, 2001.
- [25] R. Kalman, "Kronecker invariants and feedback," *Ordinary Differential Equations (L. Weiss, ed.)*, Academic Press, pp. 459–471, 1972.
- [26] A. Isidori, *Nonlinear Control Systems: An Introduction, 3rd Ed.* New-York: Springer-Verlag, 1995.
- [27] P. Rouchon, "Necessary condition and genericity of dynamic feedback linearization," *Journal Math. Syst. Estimation Control*, vol. 4, no. 2, 1994.
- [28] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot, "Representation for the radical of a finitely generated differential ideal," *International Symposium on Symbolic and Algebraic Computation*, pp. 158–166, 1995.
- [29] B. Vincke and A. Lambert, "Experimental comparison of bounded-error state estimation and constraints propagation," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 4724–4729.
- [30] V. Drevelle and P. Bonnifait, "Reliable positioning domain computation for urban navigation," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 21–29, 2013.
- [31] B. Desrochers and L. Jaulin, "A minimal contractor for the polar equation; application to robot localization," *Engineering Applications of Artificial Intelligence*, 2016.
- [32] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. M. Veres, "Reliable non-linear state estimation involving time uncertainties," *Automatica*, vol. 93, pp. 379–388, 2018.
- [33] C. Carbonnel, G. Trombettoni, P. Vismara, and G. Chabert, "Q-intersection algorithms for constraint-based robust parameter estimation," P. of AAAI, Ed. AAAI Press, 2014, pp. 2630–2636.
- [34] S. Rohou, B. Desrochers, et al., "The Codac library – Constraint-programming for robotics," 2022, <http://codac.io>.
- [35] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget, "Revising Hull and Box Consistency," in *ICLP*, 1999, pp. 230–244.
- [36] A. Tarski, "A lattice-theoretical fixpoint theorem and its applications," *Pacific Journal of Mathematics*, vol. 5, no. 2, pp. 285–309, 1955.
- [37] K. Apt, *Principles of constraint programming*. Cambridge University Press, 2003.
- [38] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. M. Veres, "Guaranteed computation of robot trajectories," *Robotics and Autonomous Systems*, vol. 93, pp. 76–84, 2017.
- [39] E. Walter and H. Piet-Lahanier, "Exact recursive polyhedral description of the feasible parameter set for bounded-error models," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 911–915, 1989.
- [40] J. Wan, S. Sharma, and R. Sutton, "Guaranteed state estimation for nonlinear discrete-time systems via indirectly implemented polytopic set computation," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4317–4322, 2018.
- [41] S. Rohou and L. Jaulin, "Exact bounded-error continuous-time linear state estimator," *Systems & Control Letters*, vol. 153, p. 104951, 2021.
- [42] J. Xiong, C. Jaubertie, L. Travé-Massuyes, and F. L. Gall, "Fault detection using interval kalman filtering enhanced by constraint propagation," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 490–495.
- [43] M. Mustafa, A. Stancu, N. Delanoue, and E. Codres, "Guaranteed SLAM; An Interval Approach," *Robotics and Autonomous Systems*, vol. 100, pp. 160–170, 2018.
- [44] C. Combastel, "A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes," in *Proceedings of CDC-ECC'05*, 2005.
- [45] L. Chisci, A. Garulli, and G. Zappa, "Recursive state bounding by parallelotopes," *Automatica*, vol. 32, no. 7, pp. 1049–1055, 1996.
- [46] A. Rauh, S. Rohou, and L. Jaulin, "An ellipsoidal predictor-corrector state estimation scheme for linear continuous-time systems with bounded parameters and bounded measurement errors," *Frontiers in Control Engineering*, vol. 3, 2022. [Online]. Available: <https://www.frontiersin.org/article/10.3389/ftceg.2022.785795>
- [47] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.
- [48] Z. Wang, A. Lambert, and X. Zhang, "Dynamic icsp graph optimization approach for car-like robot localization in outdoor environments," *Computers*, vol. 8, no. 63, 2019.
- [49] E. Colle and Galerne, "Mobile robot localization by multiangulation using set inversion," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 39–48, 2013.
- [50] Z. Wang, "Guaranteed localization and mapping for autonomous vehicles," Ph.D. dissertation, 2018, thèse de doctorat dirigée par Lambert, Alain Robotique Paris Saclay 2018. [Online]. Available: <http://www.theses.fr/2018SACL395>
- [51] M. Janssen, P. Van Hentenryck, and Y. Deville, "Optimal Pruning in Parametric Differential Equations," in *Principles and Practice of Constraint Programming - CP 2001: 7th International Conference, CP 2001 Paphos, Cyprus, November 26 - December 1, 2001 Proceedings*, T. Walsh, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 539–553.
- [52] M. Kieffer, L. Jaulin, E. Walter, and D. Meizel, "Nonlinear identification based on unreliable priors and data, with application to robot localization," in *Robustness in Identification and Control*, A. Garulli, A. Tesi, and A. Vicino, Eds., vol. LNCIS 245. London, UK: Springer-Verlag, 1999, pp. 190–203.
- [53] R. Guyonneau, S. Lagrange, and L. Hardouin, "A visibility information for multi-robot localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.
- [54] J. Porta, J. Cortes, L. Ros, and F. Thomas, "A Space Decomposition Method for Path Planning of Loop Linkages," in *Proceedings of International Conference on Intelligent Robots and Systems, IROS, 2007*, pp. 1882–1888.
- [55] M. Fliess and H. Sira-Ramirez, "Control via state estimations of some nonlinear systems," in *Proc. Symp. Nonlinear Control Systems (NOLCOS 2004)*, Stuttgart, 2004.
- [56] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA.: MIT Press, 2005.
- [57] T. Raissi, N. Ramdani, and Y. Candau, "Set membership state and parameter estimation for systems described by nonlinear differential equations," *Automatica*, vol. 40, pp. 1771–1777, 2004.
- [58] A. Gning and P. Bonnifait, "Constraints propagation techniques on intervals for a guaranteed localization using redundant data," *Automatica*, vol. 42, no. 7, pp. 1167–1175, 2006.