Non-linear control under state constraints with validated trajectories for a mobile robot towing a trailer

Joris Tillet, Luc Jaulin, Fabrice Le Bars

ENSTA-Bretagne, Lab-STICC UMR CNR 6285, 2 rue François Verny, 29806 Brest, France.

Abstract—**In this paper, we propose a set-inversion approach to validate the controller of a nonlinear system that should satisfy some state constraints. We introduce the notion of *follow set* which corresponds to the set of all output vectors such that the desired dynamics can be followed without violating the state-constraints. This follow set can then be used to choose feasible trajectories that a mobile robot will be able to follow. An illustrative example with a robot towing a trailer is presented. This example is motivated by the safe control of a boat towing a marine magnetique sensor to find wrecks.**

Index Terms—**Interval analysis, Lie derivatives, mobile robots, nonlinear control, state constraints, tank-trailer.**

I. INTRODUCTION

Nonlinear control methods [15], [37], [19] have been studied for years and have found many convincing applications in robotics [4], [9], [38], [8]. When state constraints are involved, the problem is much less studied because it requires to solve nonlinear inequality constraints mixed with nonlinear differential equations. Even if planning methods [22] have provided some interesting results, the problem can still be considered as open as soon as some guarantee is required.

In this context where nonlinear differential problems are considered, interval methods have been shown to be able to provide solutions in a reliable way [39], [12], [11] and have been successful for many robotics applications such as localization [32], [5], control [25] or planning [13], [28]. In a dynamical context, the reliability is mainly due to the possibility to integrate nonlinear differential equations in a guaranteed way [36] [40] which is not possible with other approaches, to our knowledge. Large scale problems have even been solved efficiently by adding contractors [33], [2] to the interval theory. This motivated researchers to use more frequently interval approaches for nonlinear control [34], [32] [30].

In this paper, we want to combine methods coming from nonlinear control theory such as flatness [10], or feedback linearization [21], to cast the problem of finding safe trajectories (*i.e.*, that do not enter inside the forbidden zone) into a set inversion framework. The resolution of the set inversion problem can then be performed using interval analysis [17].

Set inversion is now considered as mature enough to solve efficiently real problems (see e.g., [7], [3]).

The idea of using flatness with interval methods is not new since it has been used for robust controller design [20] [14], fault detection [29], state estimation [16] or to deal rigorously with uncertainty [31]. The principle is to use flatness to transform differential constraints into analytical inequalities using Lie derivatives. Then, the resolution is performed numerically and rigorously using interval analysis tools [26].

In this paper, we consider problems with state constraints. We propose to project this set onto the set of vector outputs. This projection is shown to be possible if we get a controller that obliges the output to obey a desired dynamics. The corresponding projected set will be called *follow set* and will then be used to find trajectories that are consistent with the constraints.

Our approach will be illustrated on the *tank trailer* problem which is known to be difficult from the control point of view, but also for planning a safe trajectory [22]. This choice is motivated by the safe control of a boat towing a magnetic sensor where the validatation of the dynamic of some state constraints related to the towing cable. Other approaches of motion planning under constraints can be used to find a probable safe trajectory [27, 1], but here the goal to provide guaranteed results.

The paper is organized as follows. Section II sets up the problem in a formal way and presents the mathematical tools that will be used for the resolution. Section III introduces the tank-trailer robot and shows how to find a controller so that the output (the center of the trailer) follows the required dynamics. Section IV defines the *follow set* and shows that it can be described as a set inversion problem. It also explains how the follow set can be used to find safe trajectories. A test-case related to the safety trajectory of the tank-trailer robot for internal and external collisions is considered in Section V. Section VI concludes the paper and gives some perspectives.

II. FORMALISM

Consider a mobile robot described by the following state equations

$$\begin{cases} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) \cdot \mathbf{u} \\ \mathbf{y} & = & \mathbf{h}(\mathbf{x}) \\ \mathbf{x} \in \mathbb{X} \end{cases} \quad (1)$$

where $\mathbf{u} \in \mathbb{R}^m$ is the vector of controls (or the vector of *actuators*), $\mathbf{x} \in \mathbb{R}^n$ is the state vector and $\mathbf{y} \in \mathbb{R}^m$ is the output vector. The functions $\mathbf{f}, \mathbf{g}, \mathbf{h}$ are assumed to be smooth. The dimensions of $\mathbf{u}$ and $\mathbf{y}$ are both equal to m. All vectors depend on the continuous time t. In our context, the system is a robot and the output $\mathbf{y}$ corresponds to its position in the workspace which may be of dimension 2 or 3. The set $\mathbb{X}$ is a state constraint that should be satisfied.

We now introduce the concept of *Lie derivatives*, classical in control theory [19]. It will allow us to express any kth derivative of any output as an analytical expression of the state $\mathbf{x}$.

Lie derivatives . We have

$$\dot{\mathbf{y}} = \underbrace{\frac{d\mathbf{h}}{d\mathbf{x}}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})}_{=\mathcal{L}_{\mathbf{f}}\mathbf{h}(\mathbf{x})} + \underbrace{\frac{d\mathbf{h}}{d\mathbf{x}}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x}) \cdot \mathbf{u}}_{=\mathcal{L}_{\mathbf{g}}\mathbf{h}(\mathbf{x})}.$$

The quantity $\mathcal{L}_{\mathbf{f}}\mathbf{h}(\mathbf{x})$ is the *Lie derivative* along with $\mathbf{f}$ of $\mathbf{h}$ at $\mathbf{x}$. We can define recursively the ith order Lie derivative by

$$\mathcal{L}_{\mathbf{f}}^i\mathbf{h}(\mathbf{x}) = \mathcal{L}_{\mathbf{f}}\mathcal{L}_{\mathbf{f}}^{i-1}\mathbf{h}(\mathbf{x}) = \frac{d(\mathcal{L}_{\mathbf{f}}^{i-1}\mathbf{h})}{d\mathbf{x}}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}).$$

Relative degree. The relative degree relative for the outputs y_j, $j = 1, \dots, m$, is the smallest integer ρ_j such that

$$\mathcal{L}_{\mathbf{g}}\mathcal{L}_{\mathbf{f}}^{\rho_j - 1} h_j(\mathbf{x}) \neq 0.$$

Controllability. We want our system to follow a specific dynamic for $\mathbf{y}$, say $\dot{\mathbf{y}} = \boldsymbol{\Psi}(\mathbf{y})$. We consider the error

$$\mathbf{e} = \dot{\mathbf{y}} - \boldsymbol{\Psi}(\mathbf{y}) = \mathcal{L}_{\mathbf{f}}\mathbf{h}(\mathbf{x}) + \mathcal{L}_{\mathbf{g}}\mathbf{h}(\mathbf{x}) \cdot \mathbf{u} - \boldsymbol{\Psi}(\mathbf{h}(\mathbf{x})).$$

If the system is controllable with $\mathbf{y}$ as an output, using classical nonlinear control method, we can find a controller $\mathbf{u} = \mathbf{c}(\mathbf{x})$ such that $\mathbf{e}(t)$ converges exponentially toward $\mathbf{0}$ [19].

The following section illustrates these concepts in the tank-trailer control problem. Then, we will see in Section IV how the state constraint could be taken into account in this context.

III. TANK TRAILER CONTROL PROBLEM

A. Model

The state equations of the tank-trailer system, represented by Figure 1, are given by:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{pmatrix} = \underbrace{\begin{pmatrix} x_5 \cos x_3 \\ x_5 \sin x_3 \\ 0 \\ x_5 \sin(x_3 - x_4) \\ 0 \end{pmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ u_1 \\ 0 \\ u_2 \end{pmatrix}}_{\mathbf{g}(\mathbf{x}) \cdot \mathbf{u}} \quad (2)$$

with (x_1, x_2) the position of the tank, x_3 its heading, x_4 the heading of the trailer and x_5 the speed of the car. Note that here $\mathbf{g}(\mathbf{x})$ does not depend on $\mathbf{x}$.
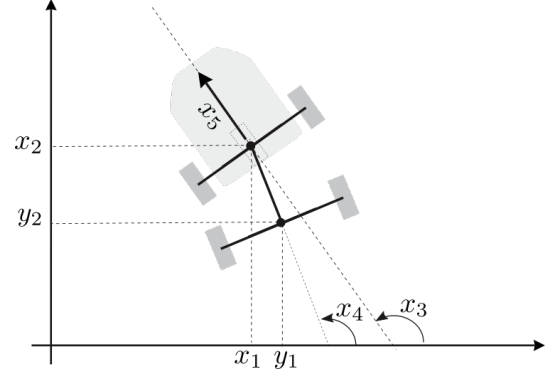

Fig. 1.   Tank with a trailer

We consider as an output the center of the trailer

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 - \cos x_4 \\ x_2 - \sin x_4 \end{pmatrix} = \mathbf{h}(\mathbf{x}). \quad (3)$$

We propose here a controller such that the output follows the desired dynamics. This choice is motivated by the control of our robot *Boatbot* which is an autonomous rubber-boat towing a kayak to which a magnetometer is attached (see Figure 2). The goal of this robot is to build a magnetic map to localize wrecks [24]. In this application, the output vector $\mathbf{y}$ corresponds to the position of the magnetometer.

Fig. 2.   Boatbot towing a magnetometer

The approach we will follow here is inspired by [35], except that here we want to follow a dynamics for $\mathbf{y}$ and not a specific trajectory.

B. Flattened feedback

The first step to applying a nonlinear control approach is to draw the graph of differential delays of the system (see Figure 3). A continuous arc corresponds to a differential delay between two variables. The dashed arc corresponds to an analytical non-differential relation relating to two variables. For instance, since we read from the state equations that $\dot{x}_1 = x_5 \cos x_3$ we add the two arcs $x_5 \to x_1$ and $x_3 \to x_1$.

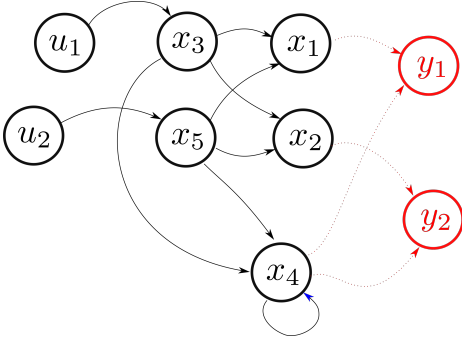Fig. 3. Graph of the differential delays of our tank-trailer vehicle

The relative degrees ρ_1, ρ_2 of y_1, y_2 can be read from the graph by counting the number of continuous arcs separating the output y_i to the inputs u_1, u_2. We get $\rho_1 = \rho_2 = 2$. Now, since $\rho_1 + \rho_2 = 2 + 2 < \dim \mathbf{x} = 5$, a feedback-based linearization method leaves a state variable without any control. If we are lucky, this floating state variable is stable and the resulting behavior is correct. Now, if we push the method up to the simulation, we observe that for our system the floating state variable is unstable. This instability makes the approach inappropriate. The following theorem, illustrated by Figure 4, provides flattened feedback of our vehicle. This means for us that the sum of the relative degrees corresponds to the dimension of the system (see [35] for details).

Theorem 1. *Consider the controller*

$$
\begin{aligned}
\dot{v}_1 &= a_1 \\
\mathbf{u} &= \underbrace{\mathbf{A}^{-1}(\mathbf{x}) \cdot \left(\begin{pmatrix} v_1 \\ a_2 \end{pmatrix} - \mathbf{b}(\mathbf{x}) \right)}_{\boldsymbol{\rho}(\mathbf{x}, v_1, a_2)}
\end{aligned} \tag{4}
$$

where

$$
\mathbf{A}(\mathbf{x}) = \begin{pmatrix} -x_5 \sin(x_3 - x_4) & \cos(x_3 - x_4) \\ x_5 \cos(x_3 - x_4) & \sin(x_3 - x_4) \end{pmatrix}
$$

and

$$
\mathbf{b}(\mathbf{x}) = \begin{pmatrix} x_5^2 \sin^2(x_3 - x_4) \\ -x_5^2 \sin(x_3 - x_4) \cos(x_3 - x_4) \end{pmatrix}.
$$

In the new coordinate system given by

$$
\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 - \cos x_4 \\ x_2 - \sin x_4 \\ x_5 \cos(x_3 - x_4) \\ v_1 \\ x_4 \\ x_5 \sin(x_3 - x_4) \end{pmatrix}}_{\boldsymbol{\varphi}(\mathbf{x}, v_1)} \tag{5}
$$

we get the closed-loop system:

$$
\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \end{pmatrix} = \begin{pmatrix} z_3 \cos z_5 \\ z_3 \sin z_5 \\ z_4 \\ a_1 \\ z_6 \\ a_2 \end{pmatrix} = \mathbf{f}_z(\mathbf{z}) + \mathbf{g}_z(\mathbf{z}) \cdot \mathbf{a}
$$

$$
\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \mathbf{h}_z(\mathbf{z})
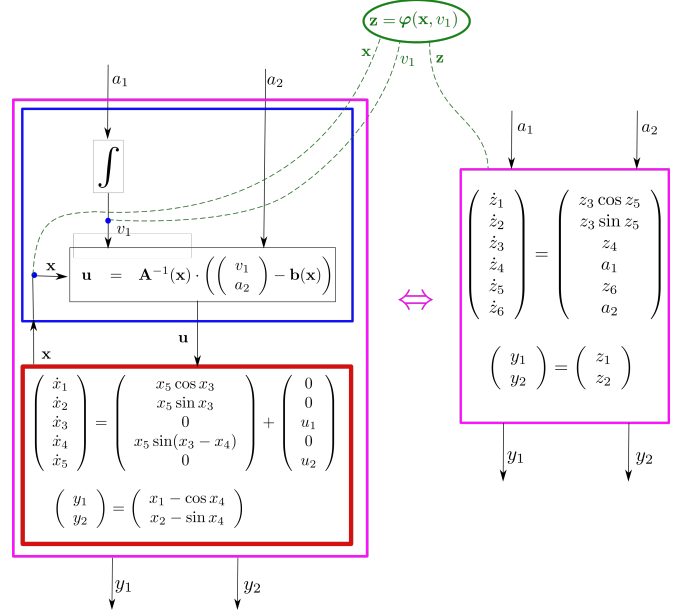$$

$$
\tag{6}
$$

Fig. 4. The two systems in the magenta boxes are equivalent

Remark 2. This theorem suggests a better coordinate system to represent the state where, (y_1, y_2) is the center of the trailer, z_5 the heading of the trailer, (z_3, z_6) the speed vector of the front car expressed in the trailer frame. It also suggest to control directly the acceleration of the trailer (via a_1) and its rotation rate (via a_2).

Proof: We have

$$\dot{\mathbf{y}} = \mathcal{L}_{\mathbf{f}}\mathbf{h}(\mathbf{x}) + \underbrace{\mathcal{L}_{\mathbf{g}}\mathbf{h}(\mathbf{x})}_{=\mathbf{0}} \cdot \mathbf{u}$$

$$= \begin{pmatrix} \dot{x}_1 + \dot{x}_4 \sin x_4 \\ \dot{x}_2 - \dot{x}_4 \cos x_4 \end{pmatrix}$$

$$= x_5 \begin{pmatrix} \cos x_3 + \sin(x_3 - x_4)\sin x_4 \\ \sin x_3 - \sin(x_3 - x_4)\cos x_4 \end{pmatrix}$$

$$= x_5 \begin{pmatrix} \cos x_3 + (\sin x_3 \cos x_4 - \cos x_3 \sin x_4)\sin x_4 \\ \sin x_3 - (\sin x_3 \cos x_4 - \cos x_3 \sin x_4)\cos x_4 \end{pmatrix}$$

$$= x_5 \begin{pmatrix} \sin x_3 \sin x_4 \cos x_4 + \cos x_3 \cos^2 x_4 \\ \sin x_3 \sin^2 x_4 - \cos x_3 \sin x_4 \cos x_4 \end{pmatrix}$$

$$= x_5 \begin{pmatrix} (\cos x_3 \cos x_4 + \sin x_3 \sin x_4)\cos x_4 \\ (\cos x_3 \cos x_4 + \sin x_3 \sin x_4)\sin x_4 \end{pmatrix}$$

$$= x_5 \cos(x_3 - x_4) \begin{pmatrix} \cos x_4 \\ \sin x_4 \end{pmatrix}$$

$$\overset{(5)}{=} z_3 \begin{pmatrix} \cos z_5 \\ \sin z_5 \end{pmatrix} \tag{7}$$

Moreover

$$\begin{cases} \dot{z}_3 = \mathcal{L}_{\mathbf{f}}z_3 + \mathcal{L}_{\mathbf{g}}z_3 \cdot \mathbf{u} \\ \ddot{z}_5 = \mathcal{L}_{\mathbf{f}}^2 z_5 + \mathcal{L}_{\mathbf{g}}\mathcal{L}_{\mathbf{f}}z_5 \cdot \mathbf{u} \end{cases}$$

or equivalently

$$\begin{pmatrix} \dot{z}_3 \\ \ddot{z}_5 \end{pmatrix} = \mathbf{A}(\mathbf{x}) \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \mathbf{b}(\mathbf{x}) \tag{8}$$

where

$$\mathbf{A}(\mathbf{x}) = \begin{pmatrix} \mathcal{L}_{\mathbf{g}_1}z_3 & \mathcal{L}_{\mathbf{g}_2}z_3 \\ \mathcal{L}_{\mathbf{g}_1}\mathcal{L}_{\mathbf{f}}z_5 & \mathcal{L}_{\mathbf{g}_2}\mathcal{L}_{\mathbf{f}}z_5 \end{pmatrix}$$

and

$$\mathbf{b}(\mathbf{x}) = \begin{pmatrix} \mathcal{L}_{\mathbf{f}}z_3 \\ \mathcal{L}_{\mathbf{f}}^2 z_5 \end{pmatrix}.$$

It is trivial to check that $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ are the matrices given in the Theorem. The matrix $\mathbf{A}(\mathbf{x})$ is singular only if $x_5 = 0$, *i.e.,* if the speed is zero. If we take the linearizing feedback $\mathbf{u} = \mathbf{A}^{-1}(\mathbf{x})\,(\mathbf{v} - \mathbf{b}(\mathbf{x}))$, then (8) becomes

$$\begin{pmatrix} \dot{z}_3 \\ \ddot{z}_5 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{9}$$

Finally

$$\dot{z}_1 \overset{(7)}{=} z_3 \cos z_5$$
$$\dot{z}_2 \overset{(7)}{=} z_3 \sin z_5$$
$$\dot{z}_3 \overset{(9)}{=} v_1$$
$$\dot{z}_4 \overset{(5)}{=} \dot{v}_1 \overset{(4)}{=} a_1$$
$$\dot{z}_5 \overset{(5)}{=} \dot{x}_4 \overset{(2)}{=} x_5 \sin(x_3 - x_4) \overset{(5)}{=} z_6$$
$$\dot{z}_6 = \ddot{z}_5 \overset{(9)}{=} v_1.$$

which corresponds to (6). ∎

As illustrated by Figure 5, the sum of the relative degrees of each output is now equal to the dimension of the system $(3 + 3 = 6)$.
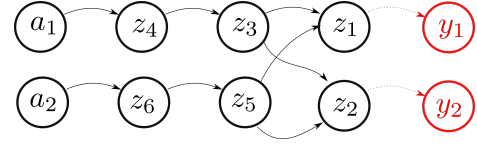

Fig. 5. Graph of the differential delays of the flattened system

C. Control the flattened system

We consider the flattened system defined by

$$\dot{\mathbf{z}} = \mathbf{f}_z(\mathbf{z}) + \mathbf{g}_z(\mathbf{z}) \cdot \mathbf{a}$$
$$\mathbf{y} = \mathbf{h}_z(\mathbf{z})$$

as defined by (6). We want $\mathbf{y}$ to follow a desired dynamics $\dot{\mathbf{y}} = \mathbf{\Psi}(\mathbf{y})$, such as, for instance, the Van der Pol equation given by:

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} y_2 \\ -(y_1^2 - 1)y_2 - y_1 \end{pmatrix}}_{\mathbf{\Psi}(\mathbf{y})}$$

The error that we want to cancel is the difference between the course of the trailer and the direction given by the vector field:

$$\begin{aligned} \mathbf{e}(\mathbf{z}) &= \dot{\mathbf{y}} - \mathbf{\Psi}(\mathbf{y}) \\ &= \mathcal{L}_{\mathbf{f}_z}\mathbf{h}_z(\mathbf{z}) - \mathbf{\Psi}(\mathbf{h}_z(\mathbf{z})) \\ &= \begin{pmatrix} z_3 \cos z_5 - z_2 \\ z_3 \sin z_5 + (0.01 \cdot z_1^2 - 1)z_2 + z_1 \end{pmatrix}. \end{aligned}$$

We have

$$\dot{\mathbf{e}}(\mathbf{z}) = \mathcal{L}_{\mathbf{f}_z}^2 \mathbf{h}_z(\mathbf{z}) - \mathcal{L}_{\mathbf{f}_z}^1 \mathbf{\Psi}(\mathbf{h}_z(\mathbf{z}))$$

$$= \begin{pmatrix} -z_3 z_6 \sin z_5 - z_3 \sin z_5 + z_4 \cos z_5 \\ (z_6 + \frac{z_1 z_2}{50} + 1)z_3 \cos z_5 + (\frac{z_1^2 z_3}{100} - z_3 + z_4)\sin z_5 \end{pmatrix}$$

and

$$\ddot{\mathbf{e}}(\mathbf{z}) = \mathcal{L}_{\mathbf{f}_z}^3 \mathbf{h}_z(\mathbf{z}) + (\mathcal{L}_{\mathbf{g}_z}L_{\mathbf{f}_z}^2 \mathbf{h}_z(\mathbf{z})) \cdot \mathbf{a} - \mathcal{L}_{\mathbf{f}_z}^2 \mathbf{\Psi}(\mathbf{h}_z(\mathbf{z}))$$

We do not give the full expressions of all quantities with respect to the z_i's for the sake of clarity. We have $\deg(e_1) = \deg(e_2) = 2$, this is why the dependency with respect to $\mathbf{a}$ occurs only at the second derivative $\ddot{\mathbf{e}}$. Let us choose the error equation

$$\ddot{\mathbf{e}} + 2\dot{\mathbf{e}} + \mathbf{e} = \mathbf{0}$$

to converge to zero. We get

$$\underbrace{\mathcal{L}_{\mathbf{f}_z}^3 \mathbf{h}_z(\mathbf{z}) + (\mathcal{L}_{\mathbf{g}_z}\mathcal{L}_{\mathbf{f}_z}^2 \mathbf{h}_z(\mathbf{z})) \cdot \mathbf{a} - \mathcal{L}_{\mathbf{f}_z}^2 \mathbf{\Psi}(\mathbf{h}_z(\mathbf{z}))}_{\ddot{\mathbf{e}}(\mathbf{z})} +$$

$$2\underbrace{(\mathcal{L}_{\mathbf{f}_z}^2 \mathbf{h}_z(\mathbf{z}) - \mathcal{L}_{\mathbf{f}_z}^1 \mathbf{\Psi}(\mathbf{h}_z(\mathbf{z})))}_{\dot{\mathbf{e}}(\mathbf{z})} +$$

$$+\underbrace{\mathcal{L}_{\mathbf{f}_z}\mathbf{h}_z(\mathbf{z}) - \mathbf{\Psi}(\mathbf{h}_z(\mathbf{z}))}_{\mathbf{e}(\mathbf{z})} = \mathbf{0}$$

or equivalently

$$
\begin{aligned}
\mathbf{a} &= \boldsymbol{\beta}(\mathbf{z}) \\
&= -\left(\mathcal{L}_{\mathbf{g}_z}\mathcal{L}_{\mathbf{f}_z}^2\,\mathbf{h}_z(\mathbf{z})\right)^{-1} \\
&\quad \cdot\left(\mathcal{L}_{\mathbf{f}_z}^3\,\mathbf{h}_z(\mathbf{z}) - \mathcal{L}_{\mathbf{f}_z}^2\,\boldsymbol{\Psi}(\mathbf{h}_z(\mathbf{z})) + 2\dot{\mathbf{e}}(\mathbf{z}) + \mathbf{e}(\mathbf{z})\right)
\end{aligned}
$$

Combining this expression with the controller (4), as illustrated by Figure 6, we get the trailer center following exactly the required vector field. Figure 7 illustrates the behavior of the controller.
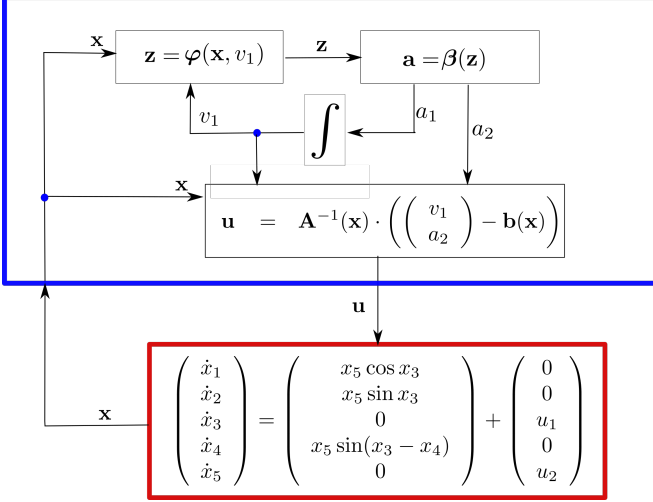

Fig. 6. With the controller, the output $\mathbf{y}$ follows exactly the Van der Pol dynamics

IV. FOLLOW SET

In the previous section, we have proposed a controller such that the output follows exactly the desired vector field. In this section, we take into account the state constraint $\mathbf{x} \in \mathbb{X}$.

Observability. A system is said to be *observable* [6] if there exist a function $\boldsymbol{\Phi}$ and integers $r_1, \ldots, r_m$ such that

$$
\mathbf{x} = \boldsymbol{\Phi}\left(y_1, \dot{y}_1, \ldots, y_1^{(r_1-1)}, \ldots, y_m, \dot{y}_m, \ldots, y_m^{(r_m-1)}\right) \tag{10}
$$

The integers r_i generally correspond to the relative degrees ρ_i for the outputs y_j, $j = 1, \ldots, m$, but this is not mandatory. This assumption is valid for many systems as if it is *flat* with the flat output $\mathbf{y}$. In what follows, we assume that we have an observable system and that $\boldsymbol{\Phi}$ is available.
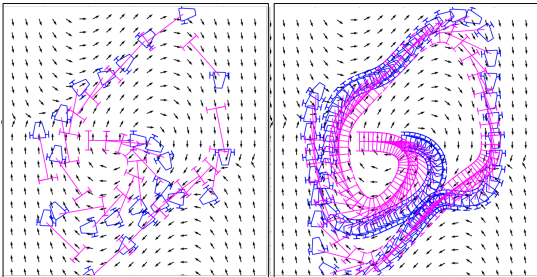

Fig. 7. Simulation of the tank (blue)-trailer (magenta) system following the Van der Pol vector field

In the case where $\mathbf{y}$ follows exactly the dynamics $\boldsymbol{\Psi}$, we can write

$$
y_j^{(i)} = \mathcal{L}_{\boldsymbol{\Psi}}^i y_j.
$$

We define the *follow set* as

$$
\mathbb{Y} = \boldsymbol{\gamma}^{-1}\left(\mathbb{X}\right),
$$

where

$$
\boldsymbol{\gamma}: \begin{array}{ccc} \mathbb{R}^m & \mapsto & \mathbb{R}^n \\ \mathbf{y} & \to & \boldsymbol{\Phi}(y_1, \mathcal{L}_{\boldsymbol{\Psi}}y_1, \ldots, \mathcal{L}_{\boldsymbol{\Psi}}^{\rho_1}y_1, \\ & & \ldots, y_m, \mathcal{L}_{\boldsymbol{\Psi}}y_m, \ldots, \mathcal{L}_{\boldsymbol{\Psi}}^{\rho_m}y_m) \end{array}
$$

and $\mathbb{X}$ is the set of state constraints that should be satisfied for the state $\mathbf{x}$. The set $\mathbb{Y}$ corresponds to the set of all $\mathbf{y}$ such that if $\mathbf{y}$ follows the dynamics $\boldsymbol{\Psi}$, then all state constraints are satisfied. Most of the time, the set $\mathbb{Y}$ cannot be computed exactly because of the non linearities between the output $\mathbf{y}$ and the state vector. So, using a set inversion approach, an inner and an outer approximations for $\mathbb{Y}$ can be obtained [18]. These approximations are computed with the SIVIA (Set Inversion Via Interval Analysis) algorithm which consists of testing interval vectors from $\mathbb{Y}$ space with a dichotomy strategy. Finally, the exact solution set is bracketed between the inner and the outer approximation.

Once this follow set has been computed, a reachability analysis could be performed to find viable domains in $\mathbf{y}$ [23]. Thus, all found $\mathbf{y}$ is such that all state constraints will always be satisfied as long as the dynamics $\boldsymbol{\Psi}$ is followed.

Consider once more the tank trailer system which is controlled so that $\mathbf{y}$ follows the dynamics $\boldsymbol{\Psi}$. If we know $y_1, y_2, \dot{y}_1, \dot{y}_2, \ddot{y}_1, \ddot{y}_2$ then we can find the corresponding $\mathbf{x}$. This is illustrated by Figure 8 where we can understand that to follow properly the desired trajectory in the $\mathbf{y}$ space, there exists a unique possibility for $\mathbf{x}(t)$. This point is clarified by the following proposition.
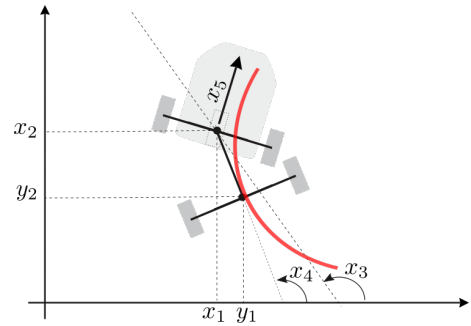

Fig. 8. From $\mathbf{y}$ and its derivatives, we can find $\mathbf{x}$

Proposition 3. *For the tank-trailer system, we have* $\mathbf{x} = \boldsymbol{\Phi}\left(y_1, \dot{y}_1, \ddot{y}_1, y_2, \dot{y}_2, \ddot{y}_2\right)$ *as given by Figure 9.*

Proof: We have

$$
\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = L_{\mathbf{f}}(\mathbf{y}) = \begin{pmatrix} x_5\cos(x_3 - x_4)\cos x_4 \\ x_5\cos(x_3 - x_4)\sin x_4 \end{pmatrix}
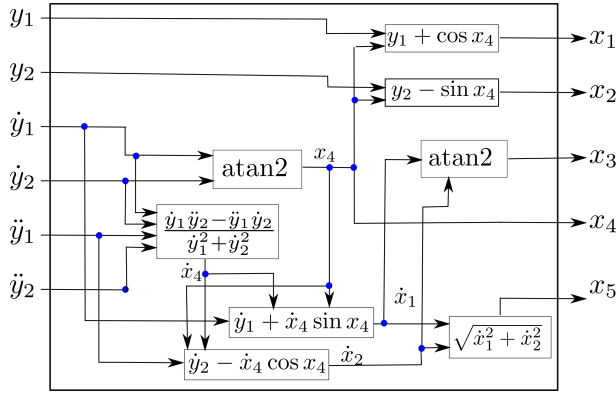$$

Fig. 9. Expression of $\Phi(y_1, \dot{y}_1, \ddot{y}_1, y_2, \dot{y}_2, \ddot{y}_2)$

Thus

$$x_4 = \text{atan2}(\dot{y}_2, \dot{y}_1) \tag{11}$$

and from (3), we get

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} - \begin{pmatrix} \cos x_4 \\ \sin x_4 \end{pmatrix}.$$

Moreover, differentiating (11), we get

$$\dot{x}_4 = \frac{\dot{y}_1 \ddot{y}_2 - \ddot{y}_1 \dot{y}_2}{\dot{y}_1^2 + \dot{y}_2^2}.$$

Thus

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} + \dot{x}_4 \begin{pmatrix} \sin x_4 \\ -\cos x_4 \end{pmatrix}$$

and

$$\begin{aligned} x_3 &= \text{atan2}(\dot{x}_2, \dot{x}_1) \\ x_5 &= \sqrt{\dot{x}_1^2 + \dot{x}_2^2} \end{aligned}$$

$\blacksquare$

V. Test-cases

In this section, we consider again the tank-trailer robot. For safety reasons, probably the most important state constraints are the collisions. Figure 10 illustrates two types of collisions: internal and external.
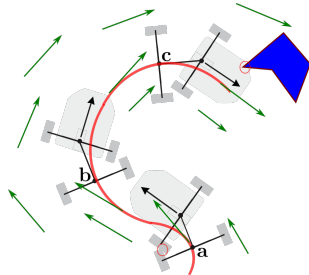

Fig. 10. Since **a** implies an internal collision and **c** corresponds to a collision with the polygon, **a**, **c** are not in the follow set $\mathbb{Y}$

External collision. Assume that there exists a polygonal obstacle to be avoided (purple in Figure 11). The orange area is the place where the trailer center cannot go safely with the required dynamics. In the green area, the output **y** can safely follow the cycle with the guarantee that the robot never

collides with the obstacle. The follow set has been computed with the algorithm SIVIA based on interval analysis. The polygonal assumption for the obstacle is not a limitation of the method. Any obstacle with a known shape could have been considered as well.
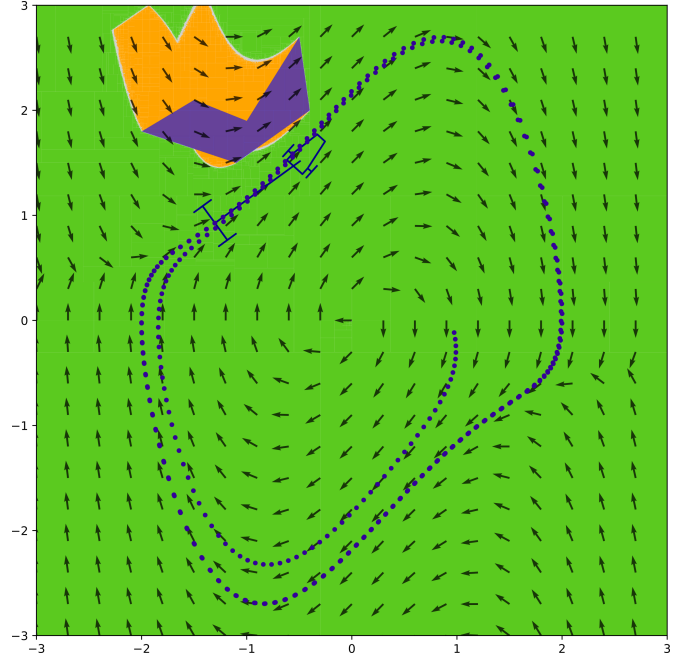

Fig. 11. The purple polygon is the obstacle the car must not collide with. The follow set is painted green. This Figure has been obtained in less than 90 seconds on a basic laptop (with a processor of 1.7GHz).

Internal collision. Assume that if a maximum angle of 70° exists between the trailer and the tank, then an internal collision occurs. The set of all acceptable trailer positions is painted green in Figure 12, assuming that the robot follows the required dynamics. Again, we have validated the painted trajectory and the limit cycle since it remains in the follow set. An illustrating video is given at the following link: https://youtu.be/xFpK147UCvE.
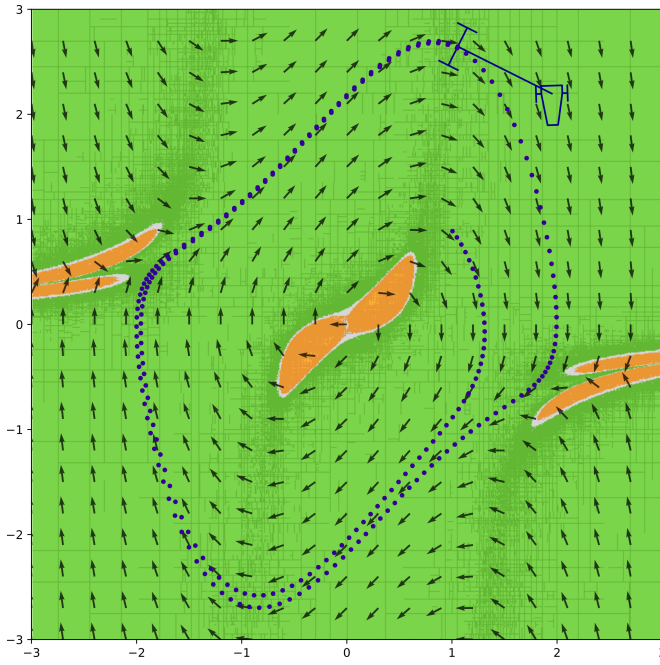
Fig. 12. The green area corresponds to the follow set associated with an internal collision. In the orange area, the controller will lead to a state with an angle $x_4 - x_3$ too strong and the non-collision constraint will be violated. This Figure has been obtained in 2 minutes.

The computed follow set is an inner approximation (and the orange area the complementary of the outer approximation), so it is a little smaller than the exact one, to guarantee a safe behavior.

VI. Conclusion

In this paper, we have considered the problem of the validation of a trajectory of a robot with its controller when state constraints exist. We have shown that using Lie derivatives, it was possible to derive non-differential constraints defining the follow set that takes place in the set of outputs **y**. Now, the dimension of **y** is usually small (typically 2 or 3) in mobile robotics, since it corresponds to the world space. As a consequence, we were able to propose an interval-based algorithm to compute an inner and an outer approximation of the follow set. If the output trajectory remains inside this set, then the trajectory can be considered as validated. An example related to the tank-trailer robot has been treated in the case where internal and external collisions should be avoided.

This problem is of particular importance in the case of articulated cumbersome robots such as marine robots where sensors have to be towed in an environment with many obstacles including rocks, islands, or other boats.

In addition to maritime applications where towing sensors such as sonars or CTD (Conductivity Temperature Depth) probes is quite typical for oceanography and hydrography, as shown in the new attached video, delivery with ground robots towing trailers full of postal packages, food deliveries, etc. could be another example of application. Since around 5 years, deliveries with aerial drones has been a quite popular topic, yet it seems at the moment there are several obstacles (such as safety, highly changing law regulations about aerial drones, energy efficiency, acceptation by people, etc.) that prevent it to be common. Fleets of ground robots with trailers could be a good alternative as long as they are able to safely follow predefined paths where they would ask for a human operator to temporarily control them in case of an unexpected event on their known path. This could be first used inside e.g. universities campus or factories, where the environment of the robot can be more easily controlled than directly in the street.

References

[1] A. Berning, A. Girard, I. Kolmanovsky, and S. D'Souza. Rapid uncertainty propagation and chance-constrained path planning for small unmanned aerial vehicles. *Advanced Control for Applications: Engineering and Industrial Systems*, 2019.

[2] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173:1079–1100, 2009.

[3] E. Colle and S. Galerne. Mobile robot localization by multiangulation using set inversion. *Robotics and Autonomous Systems*, 61(1):39–48, 2013.

[4] B. d'Andréa Novel. *Commande non-linéaire des robots*. Hermès, Paris, France, 1988.

[5] B. Desrochers, S. Lacroix, and L. Jaulin. Set-membership approach to the kidnapped robot problem. In *IROS 2015*, 2015.

[6] S. Diop and M. Fliess. Nonlinear observability, identifiability and persistent trajectories. In *Proc. 36th IEEE Conf. Decision Control*, pages 714–719, Brighton, 1991.

[7] V. Drevelle and P. Bonnifait. Localization confidence domains via set inversion on short-term trajectory. *IEEE Transactions on Robotics*, 29(5):1244–1256, oct 2013.

[8] G. El-Ghazaly, M. Gouttefarde, and V. Creuze. Hybrid cable-thruster actuated underwater vehicle-manipulator systems: A study on force capabilities. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS, Grenoble, France, 2005.

[9] I. Fantoni and R. Lozano. *Non-linear control for underactuated mechanical systems*. Springer-Verlag, 2001.

[10] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and applications. *International Journal of Control*, (61):1327–1361, 1995.

[11] 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.

[12] E. Goubault, O. Mullier, S. Putot, and M. Kieffer. Inner approximated reachability analysis. In *Proceedings of the 17th international conference on Hybrid systems: computation and control, HSCC'14*, pages 163–172, Berlin, Germany, 2014.

[13] 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.

[14] V. Hagenmeyer and E. Delaleau. Robustness analysis of exact feedforward linearization based on differential flatness. *Automatica*, (39):1941–1946, 2003.

[15] A. Isidori. *Nonlinear Control Systems: An Introduction, 3rd Ed.* Springer-Verlag, New-York, 1995.

[16] L. Jaulin. Combining interval analysis with flatness theory for state estimation of sailboat robots. *Mathematics in Computer Science*, 6(4):247–259, 2012.

[17] L. Jaulin and E. Walter. Guaranteed nonlinear estimation and robust stability analysis via set inversion. In *Proceedings of the 2nd European Control Conference*, pages 818–821, 1993.

[18] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.

[19] H.K. Khalil. *Nonlinear Systems, Third Edition*. Prentice Hall, 2002.

[20] M. Kletting, F. Antritter, and E. Hofer. Robust flatness based controller design using interval methods. In *7th IFAC Symposium on Nonlinear Control Systems*, volume 12, pages 876–881, 2007.

[21] A. J. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems and Control Letters*, 3:47–52, 1983.

[22] S. LaValle. *Planning algorithm*. Cambridge University Press, 2006.

[23] T. Le-Mézo, L. Jaulin, and B. Zerr. Bracketing the solutions of an ordinary differential equation with uncertain initial conditions. *Applied Mathematics and Computation*, 318:70–79, 2018.

[24] I. Leblond, L. Jaulin, R. Schwab, and I. Delumeau. Recherche d'objets archéologiques sous-marins à partir de données multi-capteurs. In *GRETSI'19*, 2019.

[25] N. Meslem, N. Loukkas, and J.J. Martinez. Using set invariance to design robust interval observers for discrete time linear systems. *International Journal of Robust and Nonlinear Control*, pages 1–17, 2018.

[26] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.

[27] E. Pairet, J.D. Hernandez, M. Lahijanian, and M. Carreras. Uncertainty-based online mapping and motion planning for marine robotics guidance. *IROS*, 2018.

[28] J.M. 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*, pages 1882–1888, 2007.

[29] S. Ramatou, T. Raissi, A. Zolghadri, and D. Efimov. Actuator fault detection and diagnosis for flat systems: A constraint satisfaction technique. *International Journal of Applied Mathematics and Computer Science*, 23(1), 2013.

[30] 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, may 2011.

[31] A. Rauh and E. Auer. *Modeling, Design, and Simulation of Systems with Uncertainties*. Springer, 2011.

[32] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, and S. Veres. *Reliable Robot Localization*. Wiley, dec 2019.

[33] S. Rohou, L. Jaulin, M. Mihaylova, F. Le Bars, and S. Veres. Guaranteed Computation of Robots Trajectories. *Robotics and Autonomous Systems*, 93:76–84, 2017.

[34] S. Romig, L. Jaulin, and A. Rauh. Using interval analysis to compute the invariant set of a nonlinear closed-loop control system. *Algorithms*, 12(12):262, 2019.

[35] P. Rouchon, M. Fliess, J. Levine, and P. Martin. Flatness, motion planning and trailer systems. In *Proceedings of 32nd IEEE Conference on Decision and Control*, volume 3, pages 2700–2705, Dec 1993.

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

[37] J.J. Slotine and W. Li. *Applied nonlinear control*. Prentice Hall, Englewood Cliffs (N.J.), 1991.

[38] D. Soetanto, L. Lapierre, and A. Pascoal. Adaptive, Non-Singular Path-Following Control of Dynamic Wheeled Robots. In *IEEE Conference on Decision and Control*, volume 2, pages 1765–1770, 2003.

[39] W. Tucker. A Rigorous ODE Solver and Smale's 14th Problem. *Foundations of Computational Mathematics*, 2(1):53–117, 2002.

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