Combining interval analysis with flatness theory for state estimation of sailboat robots

Luc Jaulin

Abstract. This paper proposes a new set-membership state estimator for estimating the state vector of a nonlinear dynamic robot. The method combines a symbolic technique based on flatness concepts with rigorous numerical methods based on interval analysis. Two testcases related to the state estimation of a sailboat robot are proposed to illustrate the principle and the efficiency of the approach.

Keywords. bounded-error, constraint propagation, flatness, nonlinear observers, interval analysis, sailboat, robotics, set theory.

ENSTA, LabSTICC, 2 rue François Verny
29806 Brest, France
Tel. +33 (0)2 98 34 89 10
web: `www.ensta-bretagne.fr/jaulin/`

1. Introduction

This paper presents a new interval approach for nonlinear state estimation with an application to sailboat robotics. This problem is motivated by the *microtransat challenge* where small autonomous sailboat robots are designed to cross the Atlantic ocean [5]. All components of such robots should be as robust as possible with respect to all situations (heavy weather, waves, salt water, low level of energy, long trip, . . .). For sailboat robots, two types of sensors can be considered.

- *Reliable sensors*, which could survive under all situations. Such sensors are the GPS, the compass, the gyrometers and accelerometers. All these sensors are low energy consumers, can be enclosed inside a waterproof tank and can survive for years. The GPS gives us the position of the boat and new generation GPS can also return the speed of the boat with a good accuracy by using the Doppler effect. Since magnetic perturbations inside the ocean can be neglected, the compass measures the north direction with a rather good accuracy. The gyrometer returns the rotational speed and the accelerometers provide the roll and pitch of the robot.
- *Unreliable sensors*, which have a high probability to brake down in case of heavy weather. Anemometers (a device for measuring the wind speed), weather vane (to return the direction of the wind), dynamometers which measure the forces on the sail or the rudder are considered as unreliable. They are directly in contact with aggressive natural elements (wind, wave, salt) and can fail at any time.

On the one hand, to control the robot, it is necessary to know where the wind comes from, its power and the strength of the forces on the sail or on the rudder, if the mainsheet is tight or not, . . .

(see e.g. [34], [19]). On the other hand, a reliable boat should only enclose reliable sensors. The aim of this paper is twofold.

- The first goal is to show that the variables that could be measured by the unreliable sensors could be reconstructed dynamically from the data collected by the reliable sensors. This is new in a sailboat context, even if the possibility to control a sailboat robot without any wind sensor has already been demonstrated in [38].
- The second goal is to give a new method which combines nonlinear symbolic observation techniques [9], based on flatness concepts, with interval analysis [31]. The first tool makes it possible to transform the observation problem into equations that have to be solved at each point of time whereas interval analysis provides a systematic way to solve the inversion problem [27] taking into account some interval uncertainties on the measured data. Combining interval analysis with flatness has already been considered for control [16] or source separation [29], but never for state estimation.

Section 2 shows how the state estimation problem can be transformed into a set inversion problem parametrized by the time t. Basic notions on interval analysis and the set inversion algorithm are both presented in Section 3. Section 4 presents the sailboat to be considered and illustrates the procedure to be followed to transform the state estimation problem into a chain of set inversion problems. Two simulated testcases are treated on Section 5. Section 6 concludes the paper.

2. Interval flatness approach for state estimation

This section shows how, using flatness theory, a state estimation problem can be cast into a sequence of set inversion problems that have to be solved at each instant. Shortly speaking, flatness theory can be seen as a symbolic computation approach to deal easily and efficiently with specific differential equations. Consider the system described by the following state equations

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}), \end{cases} \tag{2.1}$$

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 (or *sensors*). The functions $\mathbf{f}$ and $\mathbf{g}$ are the evolution function and the observation function, respectively. They are assumed to be as smooth as needed. The dimension of $\mathbf{u}$ and that of $\mathbf{y}$ are assumed to be both equal to m. All vectors depend on the continuous time t. The system is said to be *flat* with the flat output $\mathbf{y}$ if there exist two continuous functions $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ and integers $r_1, \dots, r_n$ such that for all t, we have

$$\begin{cases} \mathbf{x} &= \boldsymbol{\phi}\left(y_1, \dot{y}_1, \dots, y_1^{(r_1-1)}, \dots\dots, y_m, \dot{y}_m, \dots, y_m^{(r_m-1)}\right) \\ \mathbf{u} &= \boldsymbol{\psi}\left(y_1, \dot{y}_1, \dots, y_1^{(r_1)}, \dots\dots, y_m, \dot{y}_m, \dots, y_m^{(r_m)}\right). \end{cases} \tag{2.2}$$

The integers r_i correspond to the relative degrees for the outputs y_j, $j = 1, \dots, m$. According to Hermann and Krener [18], a system is observable if for any pair of state vector $(\mathbf{x}_a, \mathbf{x}_b)$, $\mathbf{x}_a$ being indistinguishable from $\mathbf{x}_b$ implies $\mathbf{x}_a = \mathbf{x}_b$. Recall that a state vector $\mathbf{x}_a$ is called indistinguishable from $\mathbf{x}_b$, if for every admissible input $\mathbf{u}$, they produce the same output. Now, from (2.2), two different states cannot produce the same output. We can conclude that all systems satisfying (2.2) are observable: the function $\boldsymbol{\phi}$ gives us the unique state vector which is consistent with the outputs and their derivatives. Of course, we assume here that the output vector $\mathbf{y}$ is measured and that we can estimate its derivatives with a good accuracy. In practice, the functions $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ involved in (2.2) are unknown. To get them, we have to proceed in two steps.

- The *derivation step* (see [20]) computes symbolically $y_1, \dot{y}_1, \dots, y_1^{(r_1)}, \dots, y_m, \dot{y}_m, \dots, y_m^{(r_m)}$ as functions of $\mathbf{x}$ and $\mathbf{u}$, using (2.1). We get an expression of the form
$$\begin{pmatrix} y_1 \\ \dot{y}_1 \\ \vdots \\ y_m^{(r_m)} \end{pmatrix} = \mathbf{h}\begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}. \tag{2.3}$$
This can be done automatically without any difficulty using symbolic computation. It suffices to take all m equations $y_j = g_j(\mathbf{x})$ and to compute symbolically its first, second, $\dots$ r_jth derivatives with respect to t. At each step, the $\dot{x}_i$ are replaced by $f_i(\mathbf{x}, \mathbf{u})$.
- The *resolution step* inverses symbolically the function $\mathbf{h}$ to get an expression of the form (2.2). This operation is difficult to obtain except for simple systems.

Example 1. Consider the system
$$\begin{cases} \dot{x}_1 &= x_1 + x_2 \\ \dot{x}_2 &= x_2^2 + u \\ y &= x_1. \end{cases}$$
For the derivation step, we compute $y, \dot{y}, \ddot{y}$ with respect to $\mathbf{x}$ and u. We get
$$\begin{cases} y &= x_1 \\ \dot{y} &= \dot{x}_1 = x_1 + x_2 \\ \ddot{y} &= \dot{x}_1 + \dot{x}_2 = x_1 + x_2 + x_2^2 + u. \end{cases}$$
Thus
$$\mathbf{h}\begin{pmatrix} \mathbf{x} \\ u \end{pmatrix} = \begin{pmatrix} x_1 \\ x_1 + x_2 \\ x_1 + x_2 + x_2^2 + u \end{pmatrix}.$$
For the resolution step, we have to isolate $\mathbf{x}, u$ to get an expression with respect to $y, \dot{y}, \ddot{y}$. We get
$$\begin{cases} x_1 &= y \\ x_2 &= \dot{y} - x_1 = \dot{y} - y \\ u &= \ddot{y} - \left(x_1 + x_2 + x_2^2\right) = \ddot{y} - \dot{y} - (\dot{y} - y)^2. \end{cases}$$
As a consequence,
$$\begin{cases} \boldsymbol{\phi}(y, \dot{y}) &= \begin{pmatrix} y \\ \dot{y} - y \end{pmatrix} \\ \psi(y, \dot{y}, \ddot{y}) &= \ddot{y} - \dot{y} - (\dot{y} - y)^2. \end{cases}$$
Note that here, the relative degree is $r = 2$. ■

Equation (2.3) can be rewritten as
$$\mathbf{z} = \mathbf{h}(\mathbf{w}), \tag{2.4}$$
where
$$\mathbf{z} = \left(y_1, \dot{y}_1, \dots, y_1^{(r_1)}, \dots\dots, y_m, \dot{y}_m, \dots, y_m^{(r_m)}\right) \text{ and} \tag{2.5}$$
$$\mathbf{w} = \left(\mathbf{x}^{\mathrm{T}}, \mathbf{u}^{\mathrm{T}}\right)^{\mathrm{T}}. \tag{2.6}$$

Assumption. We assume that for all variables involved in Equation (2.4), membership intervals are available [37]. These intervals can either be punctual if the value of the corresponding variable is known, small if the variable is measured with good accuracy or equal to $]-\infty, \infty[$ if nothing is known about the variable. For our state estimation problem, we have three types of variables.

- The input variables $u_j, j \in \{1, \dots, m\}$ can be assumed to be known exactly or with a good precision, i.e., the corresponding interval $[u_j]$ can be assumed to be small or punctual.
- The derivatives $y_j^{(k)}$ of the output variables $y_j, j \in \{1, \dots, m\}$, $k \in \{0, \dots, r_j\}$, are measured with a known error. The intervals $\left[y_j^{(k)}\right]$ containing $y_j^{(k)}$ can be considered as small. For robotic applications, the intervals for derivatives $y_j^{(k)}$ can often be obtained directly via derivative-based sensors (such as loch-Doppler systems, gyrometers or accelerometers). When no such sensor is available and when the signals y_j are not too noisy, a robust differentiation method (see e.g. [30]) can provide an estimate for the derivatives $y_j^{(k)}$ (but without any estimation of the error). This estimation might help the user to get intervals $\left[y_j^{(k)}\right]$, but without any reliability.
- The state variables x_i, $i \in \{1, \dots, n\}$ are considered as unknown. The corresponding intervals $[x_i]$ are thus $]-\infty, \infty[$.

Define the boxes

$$[\mathbf{w}] = \underbrace{[x_1] \times \cdots \times [x_n]}_{[\mathbf{x}]} \times \underbrace{[u_1] \times \cdots \times [u_m]}_{[\mathbf{u}]},$$

and

$$[\mathbf{z}] = [y_1] \times [\dot{y}_1] \times \cdots \times \left[y_1^{(r_1)}\right] \times \ldots \cdots \times [y_m] \times [\dot{y}_m] \times \cdots \times \left[y_m^{(r_m)}\right].$$

The posterior feasible set for $\mathbf{w}$ is

$$\begin{array}{rcl} \mathbb{W} & = & \{\mathbf{w} \in [\mathbf{w}], \exists \mathbf{z} \in [\mathbf{z}], \mathbf{z} = \mathbf{h}(\mathbf{w})\} \\ & = & [\mathbf{w}] \cap \mathbf{h}^{-1}([\mathbf{z}]). \end{array} \tag{2.7}$$

Characterizing the set $\mathbb{W}$ for a given t is thus a set inversion problem [27] which can be solved efficiently using interval analysis. Once $\mathbb{W}$ has been computed, the posterior feasible set $\mathbb{X}$ for $\mathbf{x}$ is easily obtained by a projection of $\mathbb{W}$ onto the $\mathbf{x}$-space.

Remark 1. If the system is flat, it is observable [6], [12], i.e., if the quantities $y_j^{(k)}, k \leq r_j - 1, j \in \{1, \dots, m\}$ are known without any error, then the set $\mathbb{X}(t)$ is a singleton. This is a direct consequence of the relations (2.2). In this paper, we only know intervals $\left[y_j^{(k)}\right]$ enclosing the $y_j^{(k)}$. As a consequence, the set $\mathbb{X}(t)$ generally encloses an infinite number of elements. However, its size can be small enough to allow us to find a control that fits to all state vectors inside $\mathbb{X}(t)$.

Remark 2. When the system is flat, we may already have an analytical expression for $\mathbf{h}^{-1}$ and thus interval methods are not required anymore for the inversion. Now, for our sailing boat or for many other engineering systems, the inversion cannot be done symbolically and a reliable inversion procedure, such as that provided by interval set inversion [27], is necessary.

Remark 3. When $\sum_{i=1}^{m} r_i > n$, the inversion problem is overdetermined and the functions $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are not unique. Equivalently, if $\sum_{i=1}^{m} r_i > n$, the dimension of the set to be inverted (equal to $\sum_{i=1}^{m}(r_i + 1)$) is larger than the number of unknowns (equal to $n + m$). This is a problem for most symbolic methods but not for the set inversion approach.

3. Set inversion with interval analysis

With an interval approach, a random variable $x \in \mathbb{R}$ is represented by an interval $[x]$ which encloses the support of its probability function. This representation is of course poorer than that provided by its probability density distribution, but it presents several advantages. *(i)* Since an interval with non-zero length is consistent with an infinite number of probability distribution functions, an interval representation is well adapted to represent random variables with imprecise probability density functions. *(ii)* An arithmetic can be developed for intervals, which makes it possible to deal with uncertainties in a reliable and easy way, even when strong nonlinearities occur. *(iii)* When the random variables are related by constraints (i.e., equations or inequalities) a propagation process (which will be explained later) provides an efficient polynomial algorithm that computes intervals enclosing all feasible values for the random variables. Interval analysis is used for robotics applications when strong nonlinearities are involved in the formulation of the problem. See, e.g., [26] for control, [7] and [28] for estimation and also [14] in the context of sailboat robotics.

3.1. Interval arithmetic

An interval is a closed and connected subset of $\mathbb{R}$. Consider two intervals $[x]$ and $[y]$ and an operator $\diamond \in \{+,-,\cdot,/\}$, we define $[x]\diamond[y]$ as the smallest interval which contains all feasible values for $x\diamond y$, if $x\in[x]$ and $y\in[y]$ (see [31]). For instance

$$\begin{array}{rl} [-1,3]+[2,5] & =[1,8], \\ [-1,3]\cdot[2,5] & =[-5,15], \\ [-1,3]/[2,5] & =[-\frac{1}{2},\frac{3}{2}]. \end{array}$$

If f is an elementary function such as $\sin,\cos,\dots$ we define $f([x])$ as the smallest interval which contains all feasible values for $f(x)$, if $x\in[x]$.

3.2. Contractors

Consider a constraint $\mathcal{C}$ (i.e., an equation or an inequality), some variables $x_1,x_2,\dots$ involved in $\mathcal{C}$ and prior interval domains $[x_i]$ for the x_i's. Interval arithmetic makes it possible to contract the domains $[x_i]$ without removing any feasible values for the x_i's. A contraction operator is called a *contractor.* When several constraints are involved, contractors are called sequentially, until no more significant contraction can be observed (see [3], [36], [25], for more details). The interval propagation method converges to a box which contains all solutions of our set of constraints. If this box is empty, it means that there is no solution. It can be shown that the box toward which the method converges does not depend on the order with which the contractors are applied [1], but the computing time is highly sensitive to this order. There is no optimal order in general, but in practice, one of the most efficient is called *forward-backward propagation*. It consists in writing the equation in the form $\mathbf{y}=\mathbf{h}(\mathbf{x})$. Then, using interval arithmetic, the intervals are propagated from $\mathbf{x}$ to $\mathbf{y}$ in a first step (*forward propagation*) and, in a second step, the intervals are propagated from $\mathbf{y}$ to $\mathbf{x}$ (*backward propagation*). The principle can be extended to problems involving quantifiers as shown in [33].

3.3. Algorithm for set inversion

We now present an algorithm [27] to characterize the set $\mathbb{W}=[\mathbf{w}]\cap\mathbf{h}^{-1}([\mathbf{z}])$, as required by Equation (2.7). The corresponding algorithm is given by the table below. The inputs of this algorithm are $[\mathbf{w}]$ which is a (possibly huge) box enclosing all feasible $\mathbf{w}=(\mathbf{x},\mathbf{u})$ for all t and $[\mathbf{z}]$ is the box defined as the Cartesian product of the intervals $\left[y_i^{(j)}\right]$ enclosing the outputs $y_i^{(j)}$ of our system at time t (see Equation (2.5)). The set $\mathbb{W}^+$ is a subpaving (i.e., a union of boxes) which encloses the feasible set $\mathbb{W}$.

Algorithm SIVIA(in: $[\mathbf{w}]$, $[\mathbf{z}]$, out: $\mathbb{X}^{+}$)
1 $\mathfrak{L} := \{[\mathbf{w}]\}$
2 repeat
3 pull $([\mathbf{w}], \mathfrak{L})$;
4 while the contractions are significant
5 compute $[\bar{\mathbf{w}}]$ enclosing $[\mathbf{w}] \cap \mathbf{h}^{-1}([\mathbf{z}])$
6 end repeat
7 bisect $[\bar{\mathbf{w}}]$ and push the resulting boxes into $\mathfrak{L}$
8 until all boxes of $\mathfrak{L}$ have a width smaller than ε
9 $\mathbb{W}^{+} := \cup \mathfrak{L}$.

The list $\mathfrak{L}$ contains boxes, the union of which encloses $\mathbb{W}$. It is initialized at Step 1 with the single box $[\mathbf{w}]$. At Step 2, a repeat-until loop is run until all boxes of $\mathfrak{L}$ have a width smaller than a given accuracy ε, which is chosen small enough to have a good accuracy on the result and large enough to respect the allowed computing time. At Step 3, the largest box is pulled out from the list. The forward-backward contractor is iterated at Step 4 until no more significant contraction can be observed, i.e., until the Hausdorff distance between the current box and the contracted box is smaller than a given threshold. At Step 7, the current box $[\bar{\mathbf{w}}]$ is bisected into two smaller boxes. These two boxes are pushed at the end of the queue $\mathfrak{L}$. At Step 9, the algorithm returns the subpaving $\mathbb{W}^{+}$ made by the union of all boxes stored in $\mathfrak{L}$. The properties of SIVIA (time and space complexity, convergence, . . .) have been studied in [27]. The complexity has been shown to be exponential with respect to the dimension of $\mathbf{w}$.

4. State estimator for the sailboat

4.1. Model used by the state estimator

The application to be considered in this paper is the estimation of the state of a sailboat in order to reconstruct the force and the direction of the wind. Three types of models are generally considered when dealing with robotics applications. They are listed now with an increasing degree of fidelity.

- The *model for the controller*. It should be as simple as possible (if possible linear) in order to be able to control the robot in a robust way. For instance, if one uses a proportional–integral–derivative (PID) controller to control the heading of a sailboat, the underlying model that is assumed is a second-order linear system, which behaves approximately as a sailboat (in a control point of view).
- The *model for the state estimator*. It should also be simple but should behave approximately as the actual robot. This model should take into account the nonlinearities of the robot and the nature of the noise.
- The *model for the simulator*. It should be as realistic as possible taking into account the environment (the swell, interaction with other boats, . . .), the sensors, the actuators, the communication, . . . [11], [15]. As a consequence, it is generally complex and non-deterministic.

We shall propose a simple deterministic model that will be assumed by our state estimator to describe the dynamics of the sailboat (see Figure 1). This model is given by the following state equations

$$\left\{\begin{array}{lll} \dot{x} & = & v\cos\theta + p_1 a\cos\psi \\ \dot{y} & = & v\sin\theta + p_1 a\sin\psi \\ \dot{\theta} & = & \omega \\ \dot{v} & = & \frac{f_s\sin\delta_s - f_r\sin u_1 - p_2 v^2}{p_9} \\ \dot{\omega} & = & \frac{f_s(p_6 - p_7\cos\delta_s) - p_8 f_r\cos u_1 - p_3\omega}{p_{10}} \\ \dot{a} & = & 0 \\ \dot{\psi} & = & 0 \\ f_s & = & p_4 a\sin(\theta - \psi + \delta_s) \\ f_r & = & p_5 v\sin u_1 \\ \gamma & = & \cos(\theta - \psi) + \cos(u_2) \\ \delta_s & = & \left\{\begin{array}{ll} \pi - \theta + \psi & \text{if } \gamma \leq 0 \\ \text{sign}(\sin(\theta - \psi)) \cdot u_2 & \text{otherwise} \end{array}\right. \end{array}\right. \tag{4.1}$$

where p_1 is the drift coefficient, p_2 is the tangential friction, p_3 is the angular friction, p_4 is the sail lift, p_5 is the rudder lift, p_9 is the mass of the boat and p_{10} is its mass moment of inertia. The distances p_6, p_7, p_8 are represented in Figure 1. All parameters p_i are assumed to be known exactly. The sailboat has two inputs: $u_1 = \delta_r$ is the angle between the rudder and the sailboat and $u_2 = \bar{\delta}_s$ is the maximum angle of the sail (which is limited by the length of the mainsheet). This model is similar to that described in [23], [22], except that here, *(i)* we added the direction of the wind ψ and its amplitude a as state variables and *(ii)* the control is not anymore the sail angle, but the length of the mainsheet, which is more realistic.

To apply the method proposed in Section 2, the model has to be deterministic. This is why we assumed that wind properties are piecewise constant by taking $\dot{a} = \dot{\psi} = 0$. We can also allow some small variations of the wind by replacing $\dot{a} = \dot{\psi} = 0$ by $\dot{a} = u_3$, $\dot{\psi} = u_4$, where the intervals for the two new inputs u_3, u_4 correspond to the feasible wind perturbations. The sailboat model has been chosen in order to illustrate the new state estimation approach developed in this paper. The strong nonlinearities of this model, its hybrid behavior (due to the fact that the mainsheet may be tight or not) make the estimation problem very difficult to solve using existing approaches. However, it can be easily solved by the presented approach. Now this model for the sailboat could be made more realistic by adapting the modeling tools described by Fossen in the context of marine vessel [10] to sailboats.

4.2. State estimator

The state estimator to be proposed is based on the previous model and assumes that x, y, θ, $\dot{x}$, $\dot{y}$, $\dot{\theta}$, $\ddot{x}$, $\ddot{y}$, $\ddot{\theta}$ are known with a given error. This assumption is rather realistic if our robot is equipped with a Doppler GPS and accelerometers. Otherwise, robust differentiation methods should be considered [30] to get $\dot{x}$, $\dot{y}$, $\ddot{x}$, $\ddot{y}$, $\dot{\theta}$, $\ddot{\theta}$. From the state equations of the model, it is easy to check that

$$\underbrace{\begin{pmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{pmatrix}}_{\mathbf{z}} = \mathbf{h} \underbrace{\begin{pmatrix} x \\ y \\ \theta \\ v \\ \omega \\ a \\ \psi \\ u_1 \\ u_2 \end{pmatrix}}_{\mathbf{w}},$$

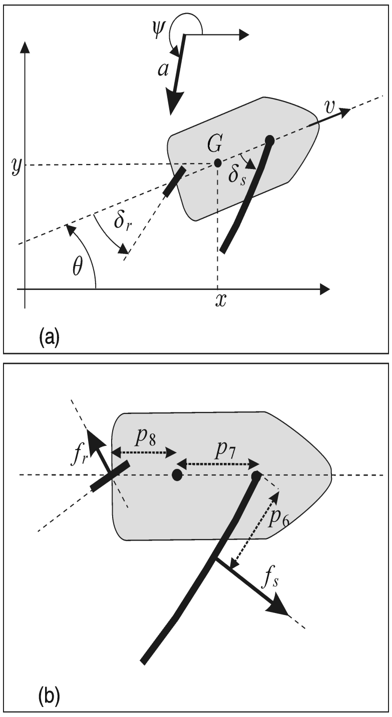

FIGURE 1. Sailboat considered to illustrate the new state estimator

where $\mathbf{h}$ is given by the following expression

$$\mathbf{h}(\mathbf{w}) = \begin{pmatrix} x \\ y \\ \theta \\ v\cos\theta + p_1 a \cos\psi \\ v\sin\theta + p_1 a\sin\psi \\ \omega \\ \frac{\left(f_s \sin\delta_s - f_r \sin u_1 - p_2 v^2\right)\cos\theta}{p_9} - \omega v \sin\theta \\ \frac{\left(f_s \sin\delta_s - f_r \sin u_1 - p_2 v^2\right)\sin\theta}{p_9} + \omega v\cos\theta \\ \frac{f_s(p_6 - p_7\cos\delta_s) - p_8 f_r \cos u_1 - p_3\omega}{p_{10}} \end{pmatrix}$$

and $f_s(\mathbf{w}), f_r(\mathbf{w}), \delta_s(\mathbf{w}), \gamma(\mathbf{w})$ are given by (4.1). From the box $[\mathbf{z}]$ enclosing the vector $\mathbf{z} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}, \ddot{x}, \ddot{y}, \ddot{\theta})^{\mathrm{T}}$, we compute the feasible set $\mathbb{W} = [\mathbf{w}] \cap \mathbf{h}^{-1}([\mathbf{z}])$. The Tchebychev center (i.e., the center of the smallest cube $\mathbb{W}$) provides us with an estimate for the state vector and thus serves as an estimation for the direction and the speed of the wind.

5. Testcases

To illustrate the behavior of our state estimator, assume that the actual robot is described by

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \boldsymbol{\varepsilon}(t) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t)), \end{cases}$$

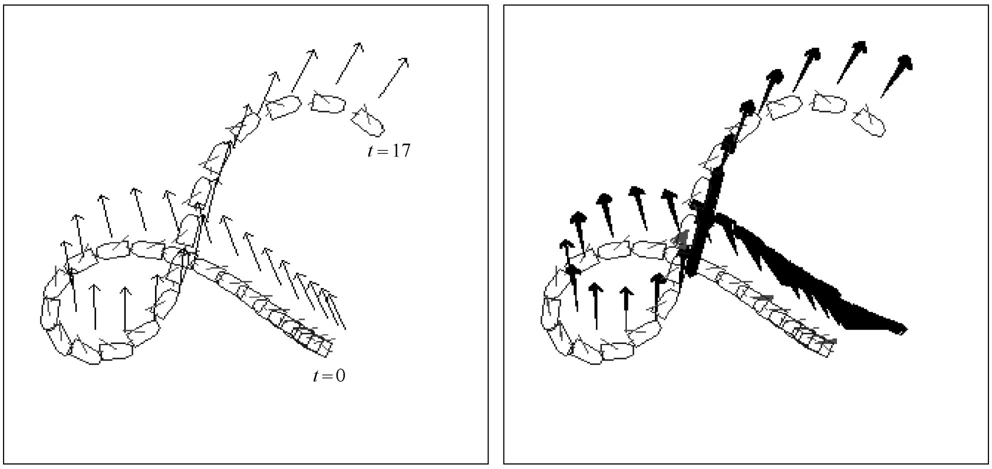

FIGURE 2. Left: simulated experiment (Testcase 1); Right: estimation of the wind

where $\boldsymbol{\varepsilon}(t)$ is the difference between the evolution used by our estimator and the actual robot. The vector $\boldsymbol{\varepsilon}(t)$ is called *model error*. It is a small quantity which encloses the model approximations, unpredictable perturbations (variations of the wind, swell, algae on the keel, . . .) or any other state noise.

5.1. Testcase 1

We consider the simulated experiment represented in Figure 2 (left). In this experiment which is started at $t_0 = 0$ and terminated at $t_{\max} = 17$ s, the boat was controlled by hand. The arrows represent the unknown wind vector, which is time dependent. For the simulation, we took

$$\begin{cases} \dot{a} = 0.2 \ \cos(0.1t) \\ \dot{\psi} = -0.1 \ \sin(0.1t) \end{cases} \tag{5.1}$$

with $a(0) = 10$ and $\psi(0) = 2$, whereas our state estimator assumes that the wind properties are piecewise constant. Thus, for this testcase, the model error is

$$\boldsymbol{\varepsilon}(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.2 \ \cos(0.1t) \\ -0.1 \ \sin(0.1t) \end{pmatrix}.$$

The parameters for the simulation have been chosen as $p_1 = 0.1, p_2 = 100$ kg·s^{-1}, $p_3 = 500$ N·m·s, $p_4 = 500$ kg·s^{-1}, $p_5 = 70$ kg·s^{-1}, $p_6 = 1.1$ m, $p_7 = 1.4$ m, $p_8 = 2$ m, $p_9 = 1000$ kg and $p_{10} = 2000$ N·m·s^2. These parameters are known by the state estimator. For all $x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}, \ddot{x}, \ddot{y}, \ddot{\theta}$ a small uniform noise inside the interval $[-2 \cdot 10^{-3}, 2 \cdot 10^{-3}]$ has been added.

The results obtained by our state estimator are represented Figure 3. At time $t_0 = 0$, the speed of the boat is small and the state estimator does not provide a good precision due to the fact that the set inversion problem is badly conditioned. At time t_1, the tuning of the sail is not optimal. As a consequence, we have two ambiguous solutions for the sail (either the sail is too closed or it is too open) which produce the same result. At time t_4 the wind come from the back and it is not possible to guess if the sail is on the right or on the left. Inside the interval $[t_2, t_3]$, we have $\gamma \leq 0$, the boat

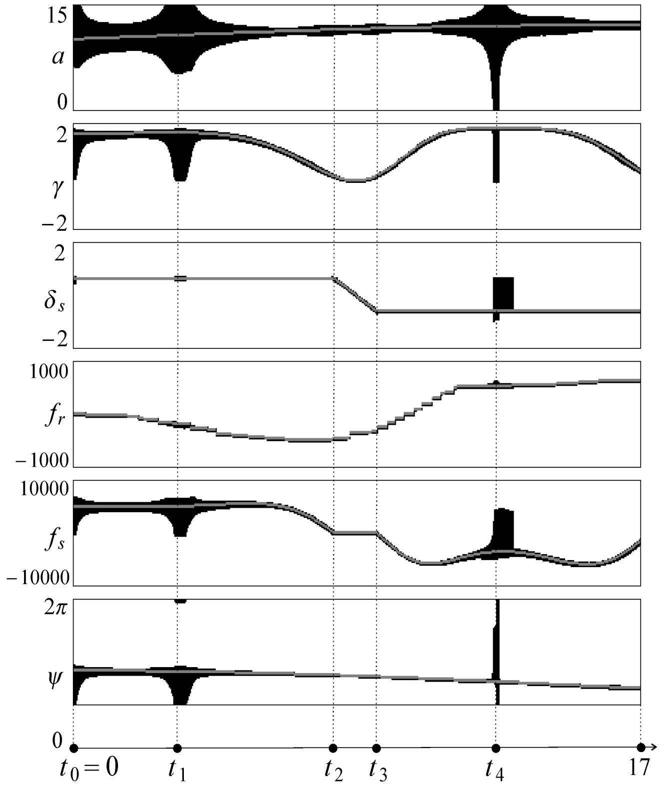

FIGURE 3. Envelopes obtained by the state estimator for Testcase 1.

is thus head to wind and the mainsheet is not tight. We checked that the interval envelopes always contain the true signals. Figure 2 (right) represents on the world frame all feasible wind vectors.

5.2. Testcase 2

We shall now consider a new simulation where the wind is still given by (5.1). However, we also added a drag force along the sail

$$f_{\text{drag}} = 30\ a\ \cos(\theta + \delta_v - \psi)$$

which slows down the robot and a swell perturbation (the waves come from East) which applies a yaw torque given by

$$T_{\text{swell}} = 30\ \sin\theta\ \cos\theta\ \cos\left(\frac{t}{10}\right).$$

As a result, the model error (unknown to our state estimator) is given by

$$\varepsilon(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{30}{p_9}\ a\ \cos(\theta + \delta_v - \psi)\ \cos\delta_v \\ \frac{30}{p_{10}}\ \sin(\theta)\ \cos(\theta)\ \cos(\frac{t}{10}) \\ 0.2\ \cos(0.1t) \\ -0.1\ \sin(0.1t) \end{pmatrix}.$$

Figure 4 (left) depicts the actual motion of the simulated sailboat with the perturbation. Note that in this figure, the tacking (turning between starboard and port tack) has been perturbed by a swell wave.

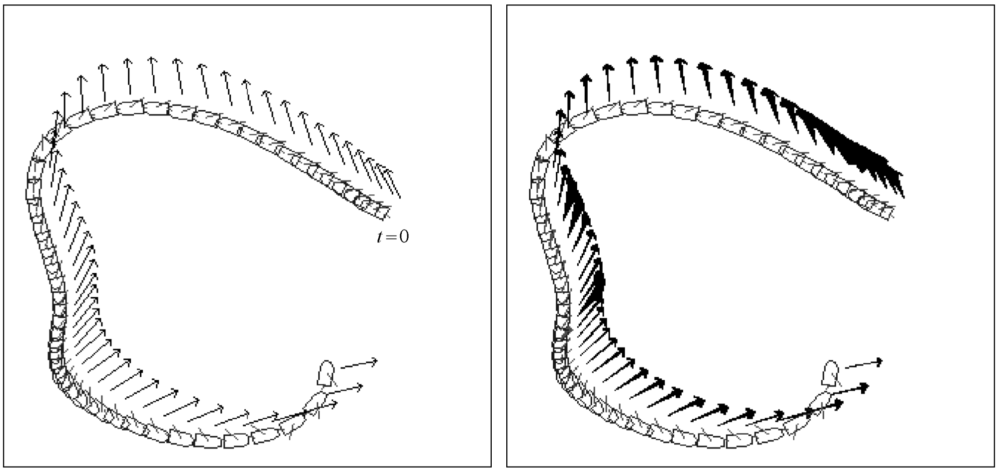

FIGURE 4. Left: simulated experiment (Testcase 2); Right: estimation of the wind

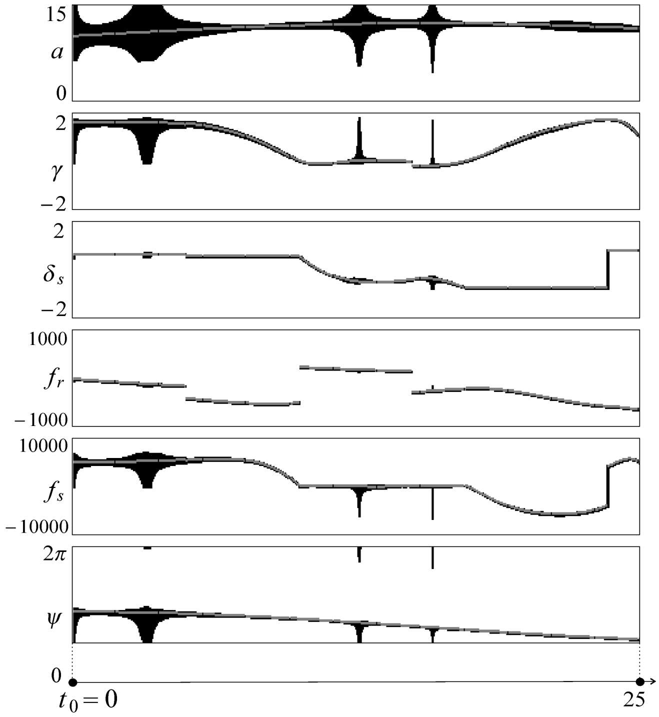

FIGURE 5. Envelopes obtained by the state estimator for Testcase 2;

The results provided by the state estimation are depicted in Figure 4 (right) and Figure 5. Most of the time, the true signals (painted grey) are inside the envelope (painted black) provided by the state estimator. When it is not the case, we observe that these true signals are close to the envelope. The fact that the envelopes do not always enclose the true signals is due to the unmodelled behaviors: the interval resolution considers that there exist no drag force and no swell perturbation.

Note that the model we have used for the simulation should be made more realistic to validate our state estimator. This could be done by building an accurate model using recent ship modelling techniques (see e.g. [4], [17], [2]).

The C++ code of the simulation as well as movies illustrating the simulated experiments with the interval state estimator can be downloaded at

`www.ensta-bretagne.fr/jaulin/getwind.html`

6. Conclusions

This paper has presented a new approach for nonlinear state estimation. This approach combines some nonlinear state estimation techniques [9] based on flatness [8] with interval set inversion. Flatness makes it possible to transform the state estimation problem in a symbolic way into set inversion problems parametrized by the time t. Interval analysis solves numerically, rigorously and efficiently the resulting set estimation problems for each t. The resulting state estimator has several advantages over classical approaches.

- The state estimator *is reliable with respect to nonlinearities*. Thanks to interval analysis, it is able to deal with nonlinear (or nondifferentiable and even noncontinuous) state equations, without linearizing (as done by the extended Kalman filter [35]) or approximating them.
- The state estimator *does not require the interval integration of differential equation*. Such integrations are needed by all other interval state estimation methods [21], [24], [32], [13] which makes them inefficient for high-dimensional systems.
- The state estimator *takes into account bounded noise* on the outputs and their derivatives. To my knowledge, it is not done by existing algebraic nonlinear state estimators.
- The state estimator *can be used for real-time applications*. For each t, interval set inversion has solved the state estimation of our sailboat problem within a time smaller than $0.1\,\mathrm{sec}$.

The approach has been illustrated on the state estimation of a sailboat. The sailboat estimation problem has several advantages: *(i)* it is motivated by the fact that we want to build a reliable boat without unreliable sensors, *(ii)* it is simple enough to illustrate the principle and the generality of presented approach and *(iii)* it is difficult enough to make all existing other deterministic nonlinear approaches for state estimation fail.

References

[1] K. Apt. The essence of constraint propagation. *Theoretical Computer Science*, 221(1-2):179–210, 1998.

[2] A. Behal, B.M. Dawson, W.E. Dixon, and Y. Fang. Tracking and regulation control of an underactuated surface vessel with nonintegrable dynamics. *IEEE Transactions on Automatic Control*, 47(3):495–500, 2002.

[3] F. Benhamou, F. Goualard, L. Granvilliers, and J-F. Puget. Revising Hull and Box Consistency. In *ICLP*, pages 230–244, 1999.

[4] V. Bertram. *Practical Ship Hydrodynamics*. F. Vieweg & Sohn, Butterworth - Heinemann, 2000.

[5] Y. Brière. *The first microtransat challenge,* `http://web.ensica.fr/microtransat`. ENSICA, 2006.

[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. High integrity gnss location zone characterization using interval analysis. In *ION GNSS*, 2009.

[8] 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.

[9] M. Fliess and H. Sira-Ramirez. Control via state estimations of some nonlinear systems. In *Proc. Symp. Nonlinear Control Systems (NOLCOS 2004)*, Stuttgart, 2004.

[10] T. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley, New York, NY, 1995.

[11] T.J. Gale and J.T. Walls. Development of a sailing dinghy simulator. *Simulation*, 74(3):167–179, 2000.

[12] J.P. Gauthier and I. Kupka. *Deterministic observation theory and applications*. Cambridge University Press, 2001.

[13] A. Gning and P. Bonnifait. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7):1167–1175, 2006.

[14] A. Goldsztejn and L. Jaulin. Inner and outer approximations of existentially quantified equality constraints. In *Proceedings of the Twelfth International Conference on Principles and Practice of Constraint Programming, (CP 2006)*, Nantes (France), 2006.

[15] G. Guillou. *Architecture multi-agents pour le pilotage automatique des voiliers de compétition et extensions algébriques des réseaux de Petri*. PhD dissertation, Université de Bretagne, Brest, France, 2011.

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

[17] H. Hansen, P.S. Jackson, and K. Hochkirch. Real-time velocity prediction program for wind tunnel testing of sailing yachts. In *International Conference on the Modern Yacht*, Southampton, 2003.

[18] R. Hermann and A. J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, October 1977.

[19] P. Herrero, L. Jaulin, J. Vehi, and M. A. Sainz. Guaranteed set-point computation with application to the control of a sailboat. *International Journal of Control Automation and Systems*, 8(1):1–7, 2010.

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

[21] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, 38:1079–1082, 2002.

[22] L. Jaulin. Modélisation et commande d'un bateau à voile. In *CIFA2004 (Conférence Internationale Francophone d'Automatique), In CDROM*, Douz (Tunisie), 2004.

[23] L. Jaulin. *Représentation d'état pour la modélisation et la commande des systèmes (Coll. Automatique de base)*. Hermès, London, 2005.

[24] L. Jaulin, M. Kieffer, I. Braems, and E. Walter. Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control*, 74(18):1772–1782, 2001.

[25] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[26] L. Jaulin, S. Ratschan, and L. Hardouin. Set computation for nonlinear control. *Reliable Computing*, 10(1):1–26, 2004.

[27] L. Jaulin and E. Walter. Guaranteed nonlinear parameter estimation via interval computations. *Interval Computation*, pages 61–75, 1993.

[28] L. Jaulin, E. Walter, O. Lévêque, and D. Meizel. Set inversion for χ-algorithms, with application to guaranteed robot localization. *Mathematics and Computers in Simulation*, 52(3-4):197–210, 2000.

[29] S. Lagrange, L. Jaulin, V. Vigneron, and C. Jutten. Nonlinear blind parameter estimation. *IEEE Transactions on Automatic Control*, 53(4):834–838, 2008.

[30] M. Mboup, C. Join, and M. Fliess. A revised look at numerical differentiation with an application to nonlinear feedback control. In *The 15th Mediterranean Conference on Control and Automation - MED'2007*, 2007.

[31] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.

[32] T. Raissi, N. Ramdani, and Y. Candau. Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40:1771–1777, 2004.

[33] S. Ratschan. Uncertainty propagation in heterogeneous algebras for approximate quantified constraint solving. *Journal of Universal Computer Science*, 6(9):861–880, 2000.

[34] C. Sauze and M. Neal. An autonomous sailing robot for ocean observation. In *proceedings of TAROS 2006*, pages 190–197, Guildford, UK, 2006.

[35] S. Thrun, W. Bugard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, M.A., 2005.

[36] M. van Emden. Algorithmic power from declarative use of redundant constraints. *Constraints*, 4(4):363–381, 1999.

[37] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, London, UK, 1997.

[38] K. Xiao, J. Sliwka, and L. Jaulin. A wind-independent control strategy for autonomous sailboats based on voronoi diagram. In *CLAWAR 2011 (best paper award)*, Paris, 2011.

Luc Jaulin
e-mail: `luc.jaulin@ensta-bretagne.fr`