

A new interval state estimator; application to sailboat robotics

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 flatness concepts with interval set inversion techniques. Two testcases related to the state estimation of a sailboat robot are proposed to illustrate the principle and the efficiency of the approach.

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

I. INTRODUCTION

This paper presents a new set-membership 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 [3]. 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 to 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 (device that is used for measuring wind speed), weather vane (which returns 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 down at any time.

On the one hand, to control the robot, it is necessary to know where the wind comes from, what is its power, how strong are the forces on the sail or on the rudder, if the mainsheet is tight or not, ... (see e.g. [23], [12]). 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.
- The second goal is to give a new method which combines nonlinear observation techniques [6], based on flatness concepts, with interval analysis [21]. The first tool makes it possible to transform the observation problem into equations that have to be solved at each time whereas interval analysis provides a systematic way to solve the inversion problem [18] taking into account some interval uncertainties on the measurement data. Note that combining interval analysis with flatness has already

been considered for control [10] or source separation [19], 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 on 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.

II. APPROACH

This section shows how a state estimation problem can be cast into a set of set inversion problems that have to be solved at each instant. Consider the system described by the following state equations

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}), \end{cases} \quad (1)$$

where $\mathbf{u} \in \mathbb{R}^m$ is the control vector, $\mathbf{x} \in \mathbb{R}^n$ is the state vector and $\mathbf{y} \in \mathbb{R}^m$ is the output vector. The dimension of \mathbf{u} and that of \mathbf{y} are assumed to both equal. All vectors depend on the continuous time t . For simplicity, we assume that the evolution function \mathbf{f} and the observation function \mathbf{g} are both smooth with respect to their arguments up to a given order r . The system is said to be *flat* with the flat output \mathbf{y} if there exist two functions ϕ and ψ such that for all t , we have

$$\begin{cases} \mathbf{x} &= \phi(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(r-1)}) \\ \mathbf{u} &= \psi(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(r-1)}, \mathbf{y}^{(r)}). \end{cases} \quad (2)$$

Now, in practice, ϕ and ψ are unknown. To get them, we have to proceed in two steps.

- The *derivation step* (see [13]) computes symbolically $\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(r)}$ with respect to \mathbf{x} and \mathbf{u} , using (1), to get an expression of the form

$$\begin{pmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \\ \vdots \\ \mathbf{y}^{(r)} \end{pmatrix} = \mathbf{h} \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}. \quad (3)$$

This can be done automatically without any difficulty using symbolic computation. It suffices to take the equation $\mathbf{y} = \mathbf{g}(\mathbf{x})$ and to compute symbolically its first, second, ... r th derivatives with respect to t . At each step, $\dot{\mathbf{x}}$ should be replaced by $\mathbf{f}(\mathbf{x}, \mathbf{u})$.

- The *resolution step* inverts symbolically the function \mathbf{h} to get an expression of the form (2). This operation is difficult to obtain except for simple systems.

Example. 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

$$\mathbf{h} \begin{pmatrix} \mathbf{x} \\ u \end{pmatrix} = \begin{pmatrix} x_1 \\ x_1 + x_2 \\ x_1 + x_2 + x_2^2 + u \end{pmatrix}.$$

Corresponding author L. Jaulin. Tel. +33 (0)2 98 34 89 10. Fax +33 (0)2 98 34 87 50. (e-mail: jaulinlu@ensta-bretagne.fr)
web:www.ensta-bretagne.fr/jaulin/
ENSTA, LabSTICC, 2 rue François Verny, 29806 Brest Cédex 09.

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

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

As a consequence,

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

Equation (3) can be rewritten as

$$\mathbf{z} = \mathbf{h}(\mathbf{w}), \quad (4)$$

where

$$\mathbf{z} = \begin{pmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \\ \vdots \\ \mathbf{y}^{(r)} \end{pmatrix} \text{ and } \mathbf{w} = \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}. \quad (5)$$

Assumption. We assume that for all variables involved in Equation (4), membership intervals are available [25]. These intervals can either be punctual if the value of the corresponding variable is known, small if the variable is measured with a 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 $u_i, i \in \{1, \dots, m\}$ can be assumed to be known exactly or with a good precision, i.e., the corresponding interval $[u_i]$ can be assumed to be small or punctual.
- The derivatives $y_i^{(j)}$ of the output variables $y_i, i \in \{1, \dots, m\}, j \in \{0, \dots, r\}$, are measured with a known error. The intervals $[y_i^{(j)}]$ containing $y_i^{(j)}$ can be considered as small. For robotic applications, the intervals for derivatives $y_i^{(j)}$ can often be obtained directly via derivative-based sensors (such as lock-Doppler system, gyrometers or accelerometers). When no such sensor is available and when \mathbf{y} is not too noisy, a robust derivation method (see e.g. [20]) can provide an estimation for the derivatives $y_i^{(j)}$ (but without any estimation of the error). This estimation might help the user to get intervals $[y_i^{(j)}]$, 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

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

and

$$[\mathbf{z}] = \underbrace{[y_1] \times \dots \times [y_m]}_{[\mathbf{y}]} \times \dots \times \underbrace{[y_1^{(r)}] \times \dots \times [y_m^{(r)}]}_{[\mathbf{y}^{(r)}]}.$$

The posterior feasible set for \mathbf{w} is

$$\begin{aligned} \mathbb{W} &= \{\mathbf{w} \in [\mathbf{w}], \exists \mathbf{z} \in [\mathbf{z}], \mathbf{z} = \mathbf{h}(\mathbf{w})\} \\ &= [\mathbf{w}] \cap \mathbf{h}^{-1}([\mathbf{z}]). \end{aligned} \quad (6)$$

Characterizing the set \mathbb{W} for a given t is thus a set inversion problem [18] 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 observable, if the vectors \mathbf{u} and $\mathbf{y}^{(j)}, j \leq r - 1$ are known without any errors, then the set $\mathbb{X}(t)$

is a singleton [4], [8]. Here, since no such assumption is done, 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. However, 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, is necessary.

III. SET INVERSION WITH INTERVAL ANALYSIS

With an interval approach, a random variable x of \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.

A. 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 [21]). For instance

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

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 $x \in [x]$.

B. 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 [17], 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 [17], 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 under 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*).

C. Algorithm for set inversion

We shall now present an algorithm [18] to characterize the set $\mathbb{W} = [\mathbf{w}] \cap \mathbf{h}^{-1}([\mathbf{z}])$, as required by Equation (6). 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 , $[\mathbf{z}]$ is the box defined as the Cartesian product of the intervals $[y_i^{(j)}]$ enclosing the outputs $y_i^{(j)}$ of our system at time t (see Equation (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 $\mathcal{L} := \{[\mathbf{w}]\}$
- 2 repeat
- 3 pull $([\mathbf{w}], \mathcal{L})$;
- 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 \mathcal{L}
- 8 until all boxes of \mathcal{L} have a width smaller than ε
- 9 $\mathbb{W}^+ := \cup \mathcal{L}$.

The list \mathcal{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 \mathcal{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 \mathcal{L} . At Step 9, the algorithm returns the subpaving \mathbb{W}^+ made by the union of all boxes stored in \mathcal{L} . The properties of SIVIA (time and space complexity, convergence, ...) have been studied in [18]. The complexity has been shown to be exponential with respect to the dimension of \mathbf{w} .

IV. STATE ESTIMATOR FOR THE SAILBOAT

A. 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 use 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, ... [7], [?]. 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 dynamic of the sailboat (see Figure 1). This model is given by following state equations

$$\begin{cases} \dot{x} = & v \cos \theta + p_1 a \cos \psi \\ \dot{y} = & v \sin \theta + p_1 a \sin \psi \\ \dot{\theta} = & \omega \\ \dot{v} = & \frac{f_s \sin \delta_s - f_r \sin u_1 - p_2 v}{p_9} \\ \dot{\omega} = & \frac{f_s (p_6 - p_7 \cos \delta_s) - p_8 f_r \cos u_1 - p_3 \omega}{p_{10}} \\ \dot{a} = & 0 \\ \dot{\psi} = & 0 \\ f_s = & p_4 a \sin (\theta - \psi + \delta_s) \\ f_r = & p_5 v \sin u_1 \\ \gamma = & \cos (\theta - \psi) + \cos (u_2) \\ \delta_s = & \begin{cases} \pi - \theta + \psi & \text{if } \gamma \leq 0 \\ \text{sign} (\sin (\theta - \psi)) . u_2 & \text{otherwise.} \end{cases} \end{cases} \quad (7)$$

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 angular inertia. The distances p_6, p_7, p_8 are represented on 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 [15], 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.

Remark 1. To apply the method proposed in Section II, the model had to be deterministic. This is why we took a constant wind $\dot{a} = \dot{\psi} = 0$, no swell, ... in the model.

Remark 2. Recall that the sailboat model is proposed in order to illustrate the new state estimation approach developed in this paper. The main advantages of this model is its strong nonlinearities, the hybrid behavior due to the fact that the mainsheet may be tight or not. This makes the estimation problem very difficult to solve using existing deterministic approaches and easily solved by the presented approach.

B. State estimator

The state estimator to be proposed is based on the previous model and assumes that $x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}, \ddot{x}, \ddot{y}, \ddot{\theta}$ 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 [20] 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

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

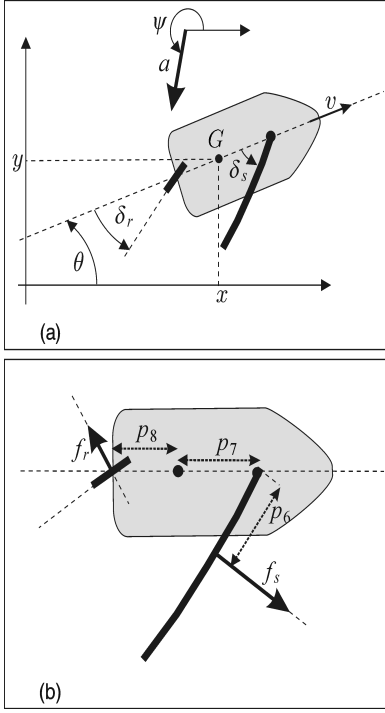


Fig. 1. Sailboat considered to illustrate the new state estimator

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

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

and $f_s(\mathbf{w})$, $f_r(\mathbf{w})$, $\delta_s(\mathbf{w})$, $\gamma(\mathbf{w})$ are given by (7). From the the box $[\mathbf{z}]$ enclosing $\mathbf{z} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}, \ddot{x}, \ddot{y}, \ddot{\theta})^T$, 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 an estimate for the state vector and thus an estimation for the direction and the speed of the wind.

V. TESTCASES

We shall now propose two testcases to illustrate the behavior of our state estimator. We shall assume that the actual robot is described by

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

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 small quantity which encloses the model approximations, unpredictable perturbations (variations of the wind, swell, algae on the keel, ...) or any other state noise.

A. Testcase 1

We consider the simulated experiment represented on Figure 2 (left). The duration this experiment is $t_{\max} = 17\text{s}$. The arrows

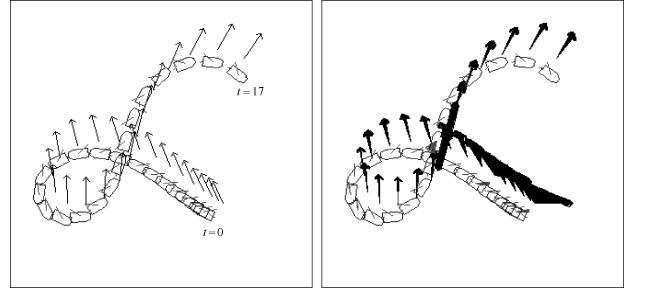


Fig. 2. Left: simulated experiment (Testcase 1); Right: estimation of the wind

represent the unknown wind vector, which is time dependent. For the simulation, we took

$$\begin{cases} \dot{a} = 0.2 \cos(0.1t) \\ \dot{\psi} = -0.1 \sin(0.1t) \end{cases} \quad (8)$$

with $a(0) = 10$ and $\psi(0) = 2$, whereas our state estimator believed that the wind is 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 $p_1 = 0.1$, $p_2 = 100 \text{ Kg.s}^{-1}$, $p_3 = 500 \text{ N.m.s}$, $p_4 = 500 \text{ Kg.s}^{-1}$, $p_5 = 70 \text{ Kg.s}^{-1}$, $p_6 = 1.1\text{m}$, $p_7 = 1.4 \text{ m}$, $p_8 = 2\text{m}$, $p_9 = 1000\text{Kg}$ and $p_{10} = 2000 \text{ N.m.s}^2$. 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. The intervals defining the frames for the variables are $[t] \in [0, 17]$, $[a] \in [0, 15]$, $[\gamma] \in [-2, 2]$, $[\psi] \in [0, 2\pi]$, $[\delta_s] \in [-2, 2]$, $[f_r] \in [-1000, 1000]$ and $[f_s] \in [-10000, 10000]$. At time $t_0 = 0$, the speed of the boat is small and the state estimator does not provide a good precision. At time t_1 , and t_4 two sail angles with different signs fit the data which yield some estimation problems. Inside the interval $[t_2, t_3]$, we have $\gamma \leq 0$, the boat 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 word frame all feasible wind vectors

B. Testcase 2

We shall now consider a new simulation where the wind is still given by (8). But we also added a drag force along the sail

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

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

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

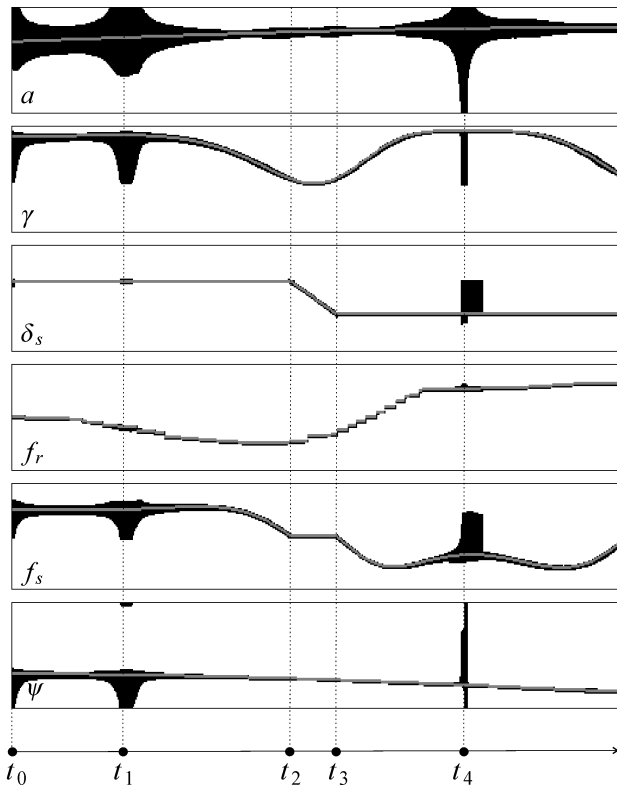


Fig. 3. Envelopes obtained by the state estimator for Testcase 1

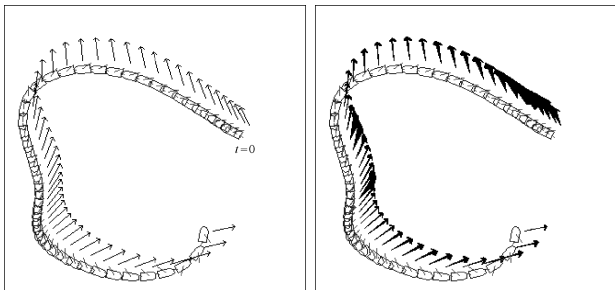


Fig. 4. Left: simulated experiment (Testcase 2); Right: estimation of the wind

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

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

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

The results provided by the state estimation are depicted on Figure 4 (right) and Figure 5. The frames are the same as for Testcase 1 except that now $t \in [0, 25]$. 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, these true signals are close to the envelope.

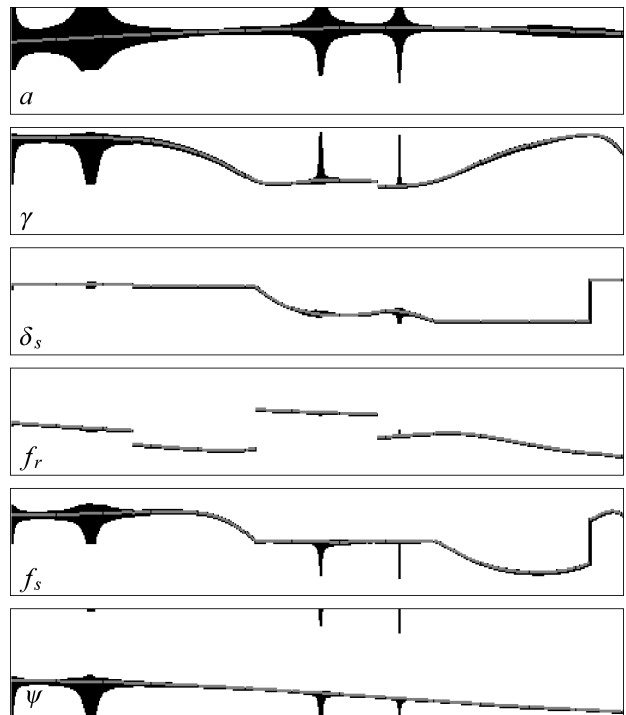


Fig. 5. Envelopes obtained by the state estimator for Testcase 2

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. [2], [11], [1]).

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

VI. CONCLUSION

This paper has presented a new approach for nonlinear state estimation. This approach combines some nonlinear state estimation techniques [6] based on flatness [5] with interval set inversion. Flatness makes possible to transform the state estimation problem into set inversion problems parametrized by time t . Interval analysis solves rigorously and efficiently the resulting set estimation problem 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 [24]) 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 [14], [16], [22], [9] which makes them inefficient for high dimensional systems.
- The state estimator *takes into account bounded noises* 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 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] 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.
- [2] V. Bertram. *Practical Ship Hydrodynamics*. F. Vieweg & Sohn, Butterworth - Heinemann, 2000.
- [3] Y. Brière. *The first microtransat challenge*, <http://web.ensica.fr/microtransat>. ENSICA, 2006.
- [4] S. Diop and M. Fliess. Nonlinear observability, identifiability and persistent trajectories. In *Proc. 36th IEEE Conf. Decision Control*, pages 714–719, Brighton, 1991.
- [5] 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.
- [6] M. Fliess and H. Sira-Ramirez. Control via state estimations of some nonlinear systems. In *Proc. Symp. Nonlinear Control Systems (NOLCOS 2004)*, Stuttgart, 2004.
- [7] T.J. Gale and J.T. Walls. Development of a sailing dinghy simulator. *Simulation*, 74(3):167–179, 2000.
- [8] J.P. Gauthier and I. Kupka. *Deterministic observation theory and applications*. Cambridge University Press, 2001.
- [9] A. Gning and P. Bonnifait. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7):1167–1175, 2006.
- [10] V. Hagenmeyer and E. Delaleau. Robustness analysis of exact feedforward linearization based on differential flatness. *Automatica*, (39):1941–1946, 2003.
- [11] 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.
- [12] 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.
- [13] A. Isidori. *Nonlinear Control Systems: An Introduction, 3rd Ed.* Springer-Verlag, New-York, 1995.
- [14] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, 38:1079–1082, 2002.
- [15] L. Jaulin. *Représentation d'état pour la modélisation et la commande des systèmes (Coll. Automatique de base)*. Hermès, London, 2005.
- [16] 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.
- [17] 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.
- [18] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.
- [19] S. Lagrange, L. Jaulin, V. Vigneron, and C. Jutten. Nonlinear blind parameter estimation. *IEEE TAC*, 53(4):834–838, 2008.
- [20] 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.
- [21] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.
- [22] 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.
- [23] C. Sauze and M. Neal. An autonomous sailing robot for ocean observation. In *in proceedings of TAROS 2006*, pages 190–197, Guildford, UK, 2006.
- [24] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, M.A., 2005.
- [25] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, London, UK, 1997.