International Journal of Control, Automation, and Systems (2010) 8(1):1-7
DOI 10.1007/s12555-010-0101-3 http://www.springer.com/12555

Guaranteed Set-point Computation with Application to the Control of a Sailboat

Pau Herrero, Luc Jaulin, Josep Vehí, and Miguel A. Sainz

Abstract: The problem of characterizing in a guaranteed way the set of all feasible set-points of a control problem is known to be difficult. In the present work, the problem to be solved involves non-linear equality constraints with variables affected by logical quantifiers. This problem is not solvable by current symbolic methods like quantifier elimination, which is commonly used for solving this class of problems. We propose the utilization of guaranteed set-computation techniques based on interval analysis, in particular a solver referred to as Quantified Set Inversion (QSI). As an application example, the problem of simultaneously controlling the speed and the orientation of a sailboat is presented. For this purpose, the combination of QSI solver and feedback linearization techniques is employed.

Keywords: Feedback linearization, interval analysis, non-linear control, set computation.

1. INTRODUCTION

Characterizing, in a guaranteed way, the set of all feasible set-points is important in many control problem (e.g., robotics). When this problem involves non-linear equality constraints with variables affected by logical quantifiers, it is known to be difficult [5]. A way to deal with this kind of problems could be quantifier elimination [21,5]. Although some applications of quantifier elimination to control systems design can be found in the literature [3,6,17], this technique is only feasible for low sized systems of polynomials inequalities. For this reason, set-computation methods based on interval analysis [4,16,19], and more specifically a solver referred to as Quantified Set Inversion (QSI) [13], is employed. An application example, consisting of the control of a sailboat, is employed in order to present the proposed approach. The outline of the paper is as follows. Section 2 introduces the application example. Section 3 presents a state space model of the sailboat. In Section 4, the polar diagram of the sailboat (i.e., set of set-points) is defined and its computation using guaranteed set computation techniques is done. In Section 5, a feedback linearization controller for the given sailboat model is designed. The required pre-compensator module, also based on guaranteed set computation techniques, is explained in Section 6. Finally, Section 7 presents some simulation results and Section 8 concludes the paper.

Manuscript received January 9, 2008; revised October 21, 2008 and March 3, 2009; accepted June 22, 2009. Recommended by Editorial Board member Kyongsu Yi under the direction of Editor Hyun Seok Yang. This work was supported by the CIBER-BBN (Network of Biomedical Research Centers in the field of Bioengineering, Biomaterials and Nanomedicine) and by Spanish government through coordinated research projects BES-2004-6337, DPI2007-668728-C02-02 and 2005-SGR-00296.

Pau Herrero is with the CIBER-BBN - Hospital de la Santa Creu i Sant Pau, 167, Sant Antoni Maria Claret, Barcelona 09025, Spain (e-mail: pherrero@santpau.cat).
Luc Jaulin is with the ENSIETA (Ecole Nationale Supérieure des Ingénieurs des Etudes et Techniques d'Armement), 2 rue François Verny, 29806 Brest Cédex 09, France (e-mail: jaulinlu@ensieta.fr).
Josep Vehí and Miguel A. Sainz are with the Institut d'Informatica i Aplicacions, Universitat de Girona, Campus de Montilivi, Edifici P4 17071 Girona, Spain (e-mails: josep.vehi@udg.edu, sainz@ima.udg.edu).

2. AUTOMATIC SAILBOAT CONTROL

To control the speed and the orientation of a sailboat, the sailor usually disposes of two actuators: the sail adjustment and the rudder adjustment. Nevertheless, these actuators are not intuitive at all and require a long training period before acquiring a good handling of the boat. For that reason, an automatic control system of the speed and orientation of a sailboat could be a priceless help for the sailor. Furthermore, many practical applications of an automatic control of the speed and the orientation of a sailboat can be found. For instance,

- During the mooring manoeuvre inside a harbor, where the speed of the sailboat has to be controlled not to surpass the speed limit.
- As a support system when crossing the oceans in solo (e.g., sleeping periods).
- For completely autonomous sailing (e.g., ecologic or military surveillance missions).

A proof of this interest is the two recent sailing competitions: the Microtransat Challenge Cup [1] held in Toulouse (France), which aims at building sailboats able to cross the Atlantic ocean in an autonomous way, and the Sailbot [2], another robotic sailboat competition held in Ontario (Canada). To our knowledge, only few works have been proposed to deal with the control of a sailboat using mathematical models [8,7,15].

2.1. Control strategy

The proposed control strategy is divided in two parts: A first one, which is executed off-line and represents the main originality of this work, aims to finding in a guaranteed way the set of admissible set-points which

© ICROS, KIEE and Springer 2010

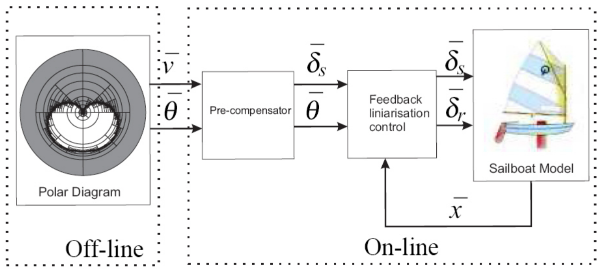


Fig. 1. Control scheme.

can be chosen by the sailor. This set corresponds to a well known sailing diagram called *polar diagram*, which is composed by the set of all pairs, the speed (v) and the orientation (θ) represented in polar coordinates, that can be reached by the sailboat in a *cruising behavior* for a given speed and orientation of the wind. In order to assure that a chosen set-point belongs to the solution set, this set is computed in a guaranteed way by means of the QSI solver, which allow to simultaneously find inner and outer approximations of the solution set.

A second part, which is executed online, consists of a controller based on feedback linearization [9]. The selection of this control technique has been motivated by the nature of the sailboat dynamics: non-linear and flat. However, as will be shown in Section 5, the feedback linearization controller does not accept the desired set-point by the user $(\bar{v},\bar{\theta})$ as linearizing outputs due to the amount of singularities that generates [14]. For this reason, a pre-compensator module is required in order to transform the set-point chosen by the user $(\bar{v},\bar{\theta})$ into a suitable set-point for the controller $(\bar{\delta}_s,\bar{\theta})$ which corresponds to the sail adjustment $\bar{\delta}_s$ and the orientation of the boat $\bar{\theta}$. Finally, the controller will compute the sail and rudder adjustments $(\bar{\delta}_s,\bar{\delta}_r)$ such that speed and orientation of the boat tends to the set-point chosen by the user. Fig. 1 graphically shows the proposed control strategy.

3. SAILBOAT MODELISATION

The sailboat represented in Fig. 2 (taken from [15]) is described by the following state equations

$$\begin{cases} \dot{x} = v\cos\theta, & \text{(i)} \\ \dot{y} = v\sin\theta - \beta v_w, & \text{(ii)} \\ \dot{\theta} = \omega, & \text{(iii)} \\ \dot{\delta}_s = u_1, & \text{(iv)} \\ \dot{\delta}_r = u_2, & \text{(v)} \\ \dot{v} = \dfrac{f_s \sin\delta_s - f_r \sin\delta_r - \alpha_f v}{m}, & \text{(vi)} \\ \dot{\omega} = \dfrac{(\ell - r_s\cos\delta_s) f_s - r_r \cos\delta_r f_r - \alpha_\theta \omega}{J}, & \text{(vii)} \\ f_s = \alpha_s (v_w \cos(\theta+\delta_s) - v\sin\delta_s), & \text{(viii)} \\ f_r = \alpha_r v \sin\delta_r, & \text{(ix)} \end{cases} \quad (1)$$

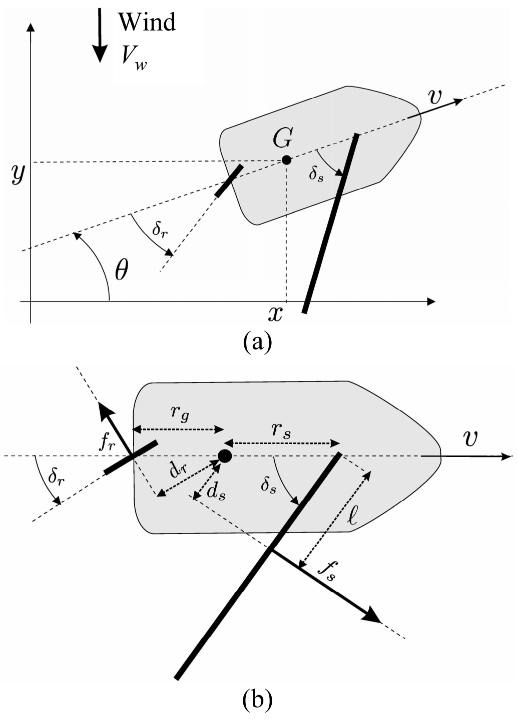


Fig. 2. Sailboat.

where $\dot{x},\dot{y},\ldots$ represents the derivatives of $x,y,\ldots$ with respect to the time t. The state vector $\mathbf{x} = (x, y, \theta, \delta_s, \delta_r, v, \omega)^{\mathrm{T}} \in \mathbb{R}^7$ is composed by

- the coordinates x, y of the inertial center G of the boat,
- the orientation θ,
- the sail angle δ_s,
- the rudder angle δ_r,
- the tangential speed v of G,
- the angular velocity ω of the boat around G.

The intermediate variables are

- the thrust force f_s of the wind on the sail,
- the force f_r of the water on the rudder.

The parameters values are

- the speed v_w $(m.s^{-1})$ of the wind,
- the distance r_r (m) between the rudder and G,
- the distance r_s (m) between the mast and G,
- the rudder lift α_r $(N.s.m^{-1})$,
- the sail lift α_s $(N.s.m^{-1})$,
- the tangential friction α_f $(N.s.m^{-1})$ of the boat with respect to the water,
- the angular friction α_θ $(N.s.m.rad^{-1})$ of the boat with respect to the water,
- the angular inertia J $(Kg.m^2)$ of the boat,
- the distance ℓ (m) between the mast and the thrust center of the sail,
- and the drift coefficient β (m^{-1}).

The parameters values are chosen as

$$\beta = 0.05,\quad r_s = 1,\quad r_r = 2,\quad \ell = 1,\quad v_w = 10,\quad m = 1000,$$
$$J = 2000,\ \alpha_f = 60,\ \alpha_\theta = 500,\ \alpha_s = 500,\ \alpha_r = 300.$$

The inputs u_1 and u_2 of the system are the derivatives of the angles δ_s and δ_r.

4. POLAR DIAGRAM

The polar diagram of the sailboat is defined by the set S of all pairs (θ, v) that can be potentially reached by the boat, in a cruising behavior.

During a cruising behavior of the boat, all state variables (except x and y) are constant,

$$\dot\theta = 0,\ \dot\delta_s = 0,\ \dot\delta_r = 0, \dot v = 0, \dot\omega = 0. \tag{2}$$

From (1), we get

$$\begin{cases} 0 &= \dfrac{f_s \sin\delta_s - f_r \sin\delta_r - \alpha_f v}{m}, \\ 0 &= \dfrac{(\ell - r_s\cos\delta_s) f_s - r_r \cos\delta_r f_r}{J}, \\ f_s &= \alpha_s\left(v_w \cos(\theta+\delta_s) - v\sin\delta_s\right), \\ f_r &= \alpha_r v \sin\delta_r, \end{cases} \tag{3}$$

which is equivalent to

$$\begin{cases} \alpha_s\left(v_w\cos(\theta+\delta_s) - v\sin\delta_s\right)\sin\delta_s - \alpha_r v \sin^2\delta_r \\ \quad -\alpha_f v = 0, \\ (\ell - r_s\cos\delta_s)\alpha_s\left(v_w\cos(\theta+\delta_s) - v\sin\delta_s\right) \\ \quad - r_r\alpha_r v\sin\delta_r\cos\delta_r = 0. \end{cases} \tag{4}$$

The polar diagram is the set $\mathbb{S}$ of all feasible vectors (v,θ) in a cruising regime, i.e.,

$$\mathbb{S} = \left\{(v,\theta) \mid \exists\delta_r, \exists\delta_s, \mathbf{f}(v,\theta,\delta_r,\delta_s) = 0\right\}, \tag{5}$$

where

$$\begin{pmatrix} \mathbf{f}(v,\theta,\delta_r,\delta_s) \\ = \alpha_s(v_w\cos(\theta+\delta_s) - v\sin\delta_s)\sin\delta_s - \alpha_r v\sin^2\delta_r \\ -\alpha_f v(\ell - r_s\cos\delta_s)\alpha_s(v_w\cos(\theta+\delta_s) - v\sin\delta_s) \\ - r_r\alpha_r v \sin\delta_r\cos\delta_r \end{pmatrix}. \tag{6}$$

4.1. Resolution

The problem stated by (5) can not be handled easily. For instance, checking whether a point (v,θ) belongs to $\mathbb{S}$ or not, requires the resolution of 2 equations with 2 unknowns, and during the regulation one needs to perform this resolution every time a new desired point is chosen. Moreover, one also may want to draw $\mathbb{S}$ to obtain a graphical representation of the polar diagram.

One possible way to handle with the problem stated by (5) is the quantifier elimination approach [21,5,17], which aims to symbolically eliminate the existential quantifier and to provide an equivalent quantifier-free representation of (5). Then, checking whether a point (v,θ) belongs to $\mathbb{S}$ becomes a trivial task. However, it is known that, although this problem is solvable, the best available algorithm has a doubly exponential complexity in the number of dimension and cannot be thought, except for toy problems. This is the reason why we have decided to use guaranteed set computation techniques, which can deal with (5) in a more efficient way (e.g., exponential complexity).

4.1.1 The QSI solver

Guaranteed set computation techniques are able to approximate, by means of sets of non-overlapping boxes (pavings), the solution set expressed by (5), instead of obtaining the exact result as quantifier elimination techniques do. Nevertheless, these approximations are most of the times sufficient from a practical point of view. It is important to remark that, in order to guarantee the pertinence of a point (v,θ) to its solution set, an inner approximation of the polar diagram is required. To our knowledge, the only available software implementation which is able to obtain such approximation is the Quantified Set Inversion (QSI) solver, which is presented in [13] and for which a practical implementation can be found in [18].

The QSI algorithm is able to deal with the following problem

$$\mathbb{S} \triangleq \left\{p \in P \mid (\exists q \in Q)\ f(p,q) = 0\right\}, \tag{7}$$

where $\mathbf{P}$ is a box of $\mathbb{R}^{n_p}$, $\mathbf{Q}$ is a box of R^{n_q} and f is a continuous function as a finite combination of elementary operators and functions such as $+,-,*,\sin$, $\cos,\ldots$ from $\mathbb{R}^n$ to $\mathbb{R}$.

The functioning of the QSI algorithm can be summarized as follows. It starts from an initial box P and checks if it belongs or not to the solution set $\mathbb{S}$ as discussed later; if P belongs to $\mathbb{S}$, it is stored into the list $\mathbb{S}^-$; if P does not belong to $\mathbb{S}$ it is rejected; if both tests are negative, the box P is bisected; the same procedure is applied recursively over the resulting boxes; the bisection procedure stops when a predefined box size ε is reached and the resulting boxes are stored into the list $\Delta\mathbb{S}$. The output of the algorithm consists of a subpaving (list of non-overlapping boxes) $\mathbb{S}^-$ representing an inner approximation of the solution set $\mathbb{S}$, and a subpaving representing all the undefined boxes $\Delta\mathbb{S}$. These subpavings provide the following bracketing of the solution set:

$$\mathbb{S}^- \subseteq \mathbb{S} \subseteq (\mathbb{S}^- \cup \Delta\mathbb{S}). \tag{8}$$

Now, the problem is reduced to prove if a box P belongs or not to $\mathbb{S}$. For this purpose, the following equivalences are used

Table 1. QSI algorithm.

```
Algorithm QSI(In: $f(\mathbf{p},\mathbf{q})=0, P_0, \varepsilon$,  Out: $\mathbb{S}^-, \Delta\mathbb{S}$)
1.  Initialization: Stack= $P_0$; $\mathbb{S}^- := \varnothing$; $\Delta\mathbb{S} := \varnothing$
2.  Repeat
3.    Unstack $P$;
4.    if $Width(P) \le \varepsilon$, ,
5.      then $\Delta\mathbb{S} := \Delta\mathbb{S} \cup P$;
6.    else if $(\forall p \in P)(\exists q \in Q) f(\mathbf{p},\mathbf{q}) = 0$ is true,
7.      then $\mathbb{S}^- := \mathbb{S}^- \cup P$;
8.    else if $(\forall p \in P)\neg((\exists q \in Q) f(\mathbf{p},\mathbf{q}) = 0)$ is true,
9.      then $P$ has no solutions;
10.   else Bisect $P$ and stack resulting boxes;
11. Until Stack= $\varnothing$;
```

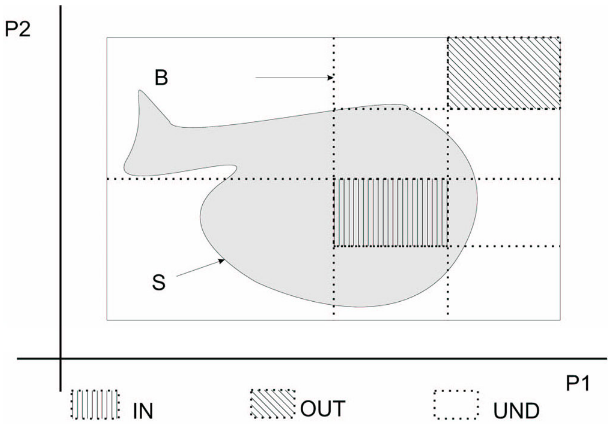


Fig. 3. Two dimensional example of the QSI algorithm.

$$\begin{aligned} P \subseteq \mathbb{S} &\Leftrightarrow (\forall p \in P)(\exists q \in Q) f(\mathbf{p},\mathbf{q}) = 0, \\ P \subseteq \neg\mathbb{S} &\Leftrightarrow (\forall p \in P)\neg((\exists q \in Q) f(\mathbf{p},\mathbf{q}) = 0). \end{aligned} \tag{9}$$

Proving the involved logical formulas in (9) is a non-trivial task. For this purpose Modal Interval Analysis (MIA) [10] is used. MIA allows proving in computationally efficient way logical formulas by transforming them into an interval inclusion test [11,20].

Table 1 shows a special instance of the QSI algorithm for the given problem. Fig. 3 shows a graphical representation of the QSI algorithm over a generic two-dimensional example in an intermediate stage of the execution. The way in which the bisection procedures progress depends on the heuristic that is chosen to select the box and the dimension to be bisected. Usually, a first-in-first-out strategy is used to select the boxes from the list and the largest interval of a box is bisected.

Given a finite precision, the termination of the QSI algorithm is guaranteed for non ill-posed problems (e.g. a small perturbation of the input involves a small modification of the solution set). As the algorithm provides inner and outer approximations to the solution set, it can be considered *sound* and *complete*.

4.1.2 Transformation of the problem

In order to pose the problem as in (7), the variable δ_r is eliminated from (5) by using symbolic calculus. Since,

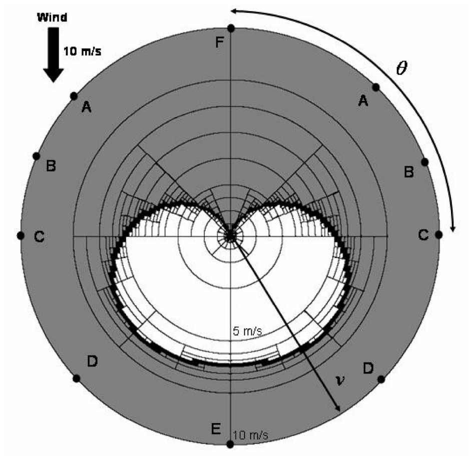


Fig. 4. Polar diagram obtained with QSI solver.

$$\sin^2\delta_r = \frac{1-\cos(2\delta_r)}{2} \text{ and } \sin\delta_r\cos\delta_r = \frac{\sin(2\delta_r)}{2}, \tag{10}$$

equation (5) can be rewritten as

$$\mathbb{S} = \{(\theta,v) \mid (\exists\delta_s \in [-\frac{\pi}{2},\frac{\pi}{2}]) f_1(\theta,v,\delta_s) = 0\}, \tag{11}$$

where $f_1(\theta,v,\delta_s)$ is given by

$$\begin{aligned} &f_1(\theta,v,\delta_s) \\ &= ((\alpha_r + 2\alpha_f)v - 2\alpha_s v_w \cos(\theta+\delta_s)\sin\delta_s + 2\alpha_s v \sin^2\delta_s)^2 \\ &+ (\frac{2\alpha_s}{r_r}(\ell - r_s\cos\delta_s)(v_w\cos(\theta+\delta_s) - v\sin\delta_s))^2 - \alpha_r^2 v^2. \end{aligned} \tag{12}$$

By using the QSI solver with a precision of $\varepsilon = 0.02$ and a fixed wind of 10 m/s, the result, obtained in less than 60 seconds on a Pentium IV, is expressed in polar coordinates in Fig. 4. The white area corresponds to the set of points (θ,v) which can be potentially reached by the sailboat, the grey area corresponds to the set of non feasible points and the black one is undefined. The wind direction and intensity should vary slowly with respect to the variations of the other state variables. Of course, if the direction of the wind changes, the polar diagram rotates. But is is more comfortable to express our state variable in a frame which is based on the wind vector. As a consequence, when the wind direction changes, we do not rotate the polar diagram, but instead, we rotate all way points.

5. FEEDBACK LINEARIZATION CONTROL

As explained in [9], feedback linearization method works if the system is flat. Many systems are flat, as the sailing boat presented in this paper. In order to apply the feedback linearization method it is necessary to differentiate the state variables, one or more times with respect to the time till the inputs u_1 or u_2 appear.

In (1), only $\dot{\delta}_s$ and $\dot{\delta}_g$ are algebraically related to $\mathbf{u}$. It is only necessary to differentiate the equations corresponding to **u**, i.e., $\dot{x}$, $\dot{y}$, $\dot{\theta}$, $\dot{v}$ and $\dot{\omega}$. We get

$$\begin{cases} \ddot{x} = \dot{v}\cos\theta - v\dot{\theta}\sin\theta, \\ \ddot{y} = \dot{v}\sin\theta + v\dot{\theta}\cos\theta, \\ \ddot{\theta} = \dot{\omega}, \\ \ddot{v} = \dfrac{\dot{f}_s\sin\delta_s + f_s u_1\cos\delta_s - \dot{f}_r\sin\delta_r - f_r u_2\cos\delta_r - \alpha_f\dot{v}}{m}, \\ \ddot{\omega} = \dfrac{u_1 r_s\sin\delta_s f_s + (\ell - r_s\cos\delta_s)\dot{f}_s}{J} \\ \quad + \dfrac{r_r(u_2\sin\delta_r f_r - \cos\delta_r\dot{f}_r) - \alpha_\theta\dot{\omega}}{J} \end{cases} \tag{13}$$

with

$$\begin{cases} \dot{f}_s = -\alpha_s v_w(\omega + u_1)\sin(\theta + \delta_s) - \alpha_s\dot{v}\sin\delta_s \\ \quad - \alpha_s v u_1\cos\delta_s, \\ \dot{f}_r = \alpha_r(\dot{v}\sin\delta_r + v u_2\cos\delta_r). \end{cases} \tag{14}$$

Notice that, as $\dot{v}$, $\dot{\theta}$, $\dot{\omega}$ are analytic functions of the state (see (1)), it is possible to consider that we have an analytic function of $\ddot{x}$, $\ddot{y}$, $\ddot{\theta}$, $\ddot{v}$, $\ddot{\omega}$ depending on the state and inputs. It is necessary to differentiate again all these quantities which do not algebraically depend on **u**, which means $\ddot{x}$, $\ddot{y}$ and $\ddot{\theta}$. Then we get

$$\begin{cases} \dddot{x} = \ddot{v}\cos\theta - 2\dot{v}\dot{\theta}\sin\theta - v\ddot{\theta}\sin\theta - v\dot{\theta}^2\cos\theta, \\ \dddot{y} = \ddot{v}\sin\theta + 2\dot{v}\dot{\theta}\cos\theta + v\ddot{\theta}\cos\theta - v\dot{\theta}^2\sin\theta, \\ \dddot{\theta} = \ddot{\omega}. \end{cases} \tag{15}$$

As the system has two inputs, it is necessary to choose two outputs in order to do the feedback linearization. A first possibility consists of choosing as outputs, the speed $y_1 = v$ and the orientation $y_2 = \theta$. This choice can be justified because **y** is a flat output for the sub-system described by the (i-vii) equations of Equation 1. However, the resulting linearizing control presents too many singularities [14]. Due to lake of space, the equation development to get these singularities has been omitted. For this reason, we have decided to choose another output which generates less singularities. Let us choose now as outputs the sail adjustment $y_1 = \delta_s$ and the orientation $y_2 = \theta$. We have

$$\begin{pmatrix} \dot{y}_1 \\ \dddot{y}_2 \end{pmatrix} = \begin{pmatrix} \dot{\delta}_s \\ \dddot{\theta} \end{pmatrix} = \mathbf{A}_1(\mathbf{x})\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \mathbf{A}_2(\mathbf{x})\begin{pmatrix} \dot{f}_s \\ \dot{f}_r \end{pmatrix} + \mathbf{b}_1(\mathbf{x}) \tag{16}$$

with

$$\mathbf{A}_1(\mathbf{x}) = \begin{pmatrix} 1 & 0 \\ \dfrac{r_s f_s\sin\delta_s}{J} & \dfrac{r_r f_r\sin\delta_r}{J} \end{pmatrix}, \quad \mathbf{A}_2(\mathbf{x}) = \begin{pmatrix} 0 & 0 \\ \dfrac{\ell - r_s\cos\delta_s}{J} & -\dfrac{r_r\cos\delta_r}{J} \end{pmatrix}, \tag{17}$$

$$\mathbf{b}_1(\mathbf{x}) = \begin{pmatrix} 0 \\ -\dfrac{\alpha_\theta\dot{\omega}}{J} \end{pmatrix},$$

where $\dot{f}_s$ and $\dot{f}_r$ are given by (14). Then, we have a relation of the form

$$\begin{aligned} \begin{pmatrix} \dot{y}_1 \\ \dddot{y}_2 \end{pmatrix} &= \mathbf{A}_1\mathbf{u} + \mathbf{A}_2(\mathbf{A}_3\mathbf{u} + \mathbf{b}_2) + \mathbf{b}_1 \\ &= (\mathbf{A}_1 + \mathbf{A}_2\mathbf{A}_3)\mathbf{u} + \mathbf{A}_2\mathbf{b}_2 + \mathbf{b}_1 \\ &= \mathbf{A}(\mathbf{x})\mathbf{u} + \mathbf{b}(\mathbf{x}), \end{aligned} \tag{18}$$

where

$$\mathbf{A}_3 = \begin{pmatrix} -\alpha_s v_w\sin(\theta + \delta_s) - \alpha_s v\cos\delta_s & 0 \\ 0 & \alpha_r v\cos\delta_r \end{pmatrix}, \tag{19}$$

and

$$\mathbf{b}_3 = \begin{pmatrix} -\alpha v_w w\sin(\theta + \delta_s) - \alpha_s\dot{v}\sin\delta_s \\ \alpha_r\dot{v}\sin\delta_r \end{pmatrix}. \tag{20}$$

In order to impose $(\dot{y}_1, \dddot{y}_2)$ to a given set-point $\mathbf{v} = (v_1, v_2)$, it is necessary to take

$$\mathbf{u} = \mathbf{A}^{-1}(\mathbf{x})(\mathbf{v} - \mathbf{b}(\mathbf{x})). \tag{21}$$

The looped system is governed by the differential equations

$$\begin{cases} \dot{y}_1 = v_1, \\ \dddot{y}_2 = v_2, \end{cases} \tag{22}$$

which are linear and decoupled. The singularities correspond to

$$\det(\mathbf{A}(\mathbf{x})) = \frac{r_r}{J}\left(f_r\sin\delta_r - v\alpha_r\cos^2\delta_r\right) = 0, \tag{23}$$

which is equivalent to

$$-v\cos^2(2\delta_r) = 0, \tag{24}$$

i.e.,

$$v = 0 \text{ or } \delta_r \in \left\{-\frac{\pi}{4}, \frac{\pi}{4}\right\}. \tag{25}$$

Such configuration corresponds to a singularity which should be avoided. As the linear systems is of 4th order instead of 7th, three out seven state variables, x, y, and v, are not controllable. The loss of control over x and y is predicable since we can drive the boat from one way point to another but, we cannotsay that we can control the location of the boat, even if we have a partial control of it. For instance, we cannot stop the boat at a given location. Hence, it is natural that it corresponds to an instability. The loss of control over v does not have any consequence because the associated dynamic is stable.

6. PRE-COMPENSATOR

As mentioned in Section 5, it is not possible to choose the speed v and the orientation θ of the sailboat as linearizing outputs for the feedback linearization controller because of the amount of singularities that provokes. Instead of that, the sail adjustment δ_s and the orientation of the sailboat θ can be chosen. As we want to control v and θ, it is necessary to introduce a pre-compensator which allows to transform the set-point chosen by the user $(\bar{v}, \bar{\theta})$ to an admissible set-point by the controller $(\delta_s, \bar{\theta})$.

Fixed $(\bar{v}, \bar{\theta})$ chosen by the user from the polar diagram, a guaranteed local search algorithm is applied for finding δ_s. This algorithm recursively bisects the initial range for the sail adjustment, Δ_s, and tests, by means of interval analysis, if a solution for the next equation

$$(\exists \delta_s \in \Delta_s) f_1(\bar{\theta}, \bar{v}, \delta_s) = 0, \tag{26}$$

exists or not in the resulting intervals. Notice that multiple solutions could exist and the one given by our algorithm is not necessarily the optimal one in terms of sailing. This bisection procedure is recursively repeated till a small enough interval is achieved for δ_s. As the controller requires a punctual value as input, the center of the resulting interval is chosen.

7. SIMULATION RESULTS

The presented control strategy has been implemented in order to show its viability. A demonstration software is available in [12]. Its use consists of selecting from the polar diagram the desired set-point by clicking with the mouse over it. Fig. 5 shows a simulation of the required manoeuver to moor the sailboat inside a harbor. Note that the boat has to go against the wind and a zigzag trajectory with sharp turns is necessary.

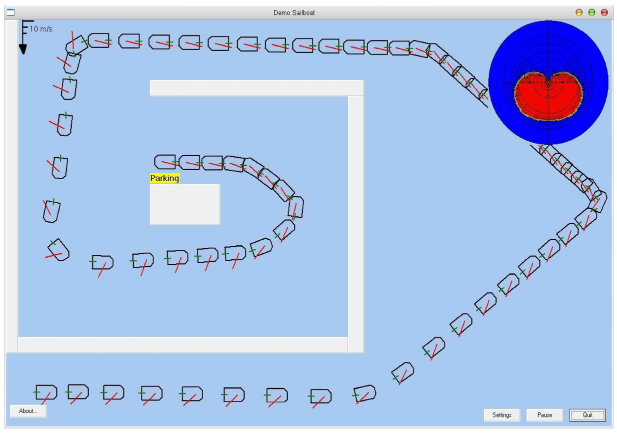


Fig. 5. Required manoeuver to moor the sailboat inside a harbor.

8. CONCLUSIONS AND FUTURE WORK

In this paper, the difficult problem of characterizing, in a guaranteed way, the set of feasible set-points of a control problem involving non-linear equality constraints with variables affected by logical quantifiers is solved for first time. To solve this problem, the utilization of approximate methods based on interval analysis, in particular a solver referred to as Quantified Set Inversion (QSI), is used. As an application, the problem of controlling the speed and the orientation of a sailboat is presented. For this purpose, the combination of the QSI solver and feedback linearization control techniques is employed. As future work, the obtention of a more precise model of a sailboat and the implementation of the proposed control strategy over a real sailboat are thought. We also plan to apply the proposed technique to other control problems.

REFERENCES

[1] *Microtransat Challenge Cup*. http://www. microtransat.org.

[2] *Sailbot Competition*. http://engsoc.queensu.ca/ sailboat/competition/index.htm.

[3] C. Abdallah, P. Dorato, W. Yang, R. Liska, and S. Steinberg, “Applications of quantifier elimination theory to control system design,” *Proc. of the 4th IEEE Mediterranean Symposium on Control and Automation, Crete, Greece*, 1996.

[4] F. Benhamou and F. Goulard, “Universally quantified interval constraints,” *Lecture Notes in Computer Science - Proc. of the 6th International Conference on Principles and Practice of Constraint Programming*, vol. 1894, pp. 67-82, 2000.

[5] G. E. Collins, “Quantifier elimination for real closed fields by cylindrical algebraic decomposition,” *Proc. of the 2nd GI Conf. Automata Theory and Formal Languages*, vol. 33, pp. 134-189. Springer, 1975.

[6] P. Dorato, “Quantified multivariate polynomial inequalities,” *IEEE Control Systems Magazine*, vol. 20, no. 5, pp. 48-58, 2000.

[7] G. Elkaim and R. Kelbley, “Control architecture for segmented trajectory following of a wind-propelled autonomous catamaran,” *Proc. of the AIAA Guidance, Navigation, and Control Conference,* Keystone, CO, 2006.

[8] G. Elkaim, B. Woodley, and R. Kelbley, “Model free subspace H-infinity control for an autonomous catamaran,” *Proc. of the ION/IEEE Position, Location, and Navigation Symposium*, San Diego, CA, 2006.

[9] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, “Flatness and defect of non-linear systems: introductory theory and examples,” *International Journal of Control*, vol. 61, no. 6, pp. 1327-1361, 1995.

[10] E. Gardeñes, M. Á. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, and A. Trepat, “Modal intervals,” *Reliable Computing*, vol. 7, no. 2, pp. 77-111, 2001.

[11] P. Herrero, *Quantified Real Constraint Solving Using Modal Intervals with Applications to Control*, Ph.D. thesis, Universitat de Girona, Girona, Spain, 2006.

[12] P. Herrero and L. Jaulin, *Software: SailBoat control using set computation and feedback linearization*, http://pau.herrero.googlepages.com/software.

[13] P. Herrero, J. Vehí, M. Á. Sainz, and L. Jaulin, "Quantified set inversion algorithm," *Reliable Computing*, vol. 11, no. 5, pp. 369 - 382, 2005.

[14] R. Hirschorn, "Output tracking through singularities," *SIAM Journal on Control and Optimization*, vol. 40, no. 1, pp. 993-1010, 2001.

[15] L. Jaulin, *Représentation d'état pour la modélisation et la commande des systèmes*, Hermes, 2001.

[16] L. Jaulin, I. Braems, and E. Walter, "Interval methods for nonlinear identification and robust control," *Proc. of Conference on Decision and Control, La Vegas*, 2002.

[17] M. Jirstrand, "Nonlinear control system design by quantifier elimination," *Journal of Symbolic Computation*, vol. 24, no. 2, pp. 137-152, 1997.

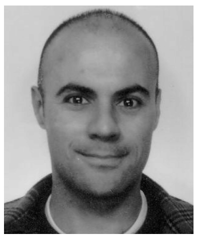
[18] MiceLab, *MISO: Modal Interval Software*, http://pau.herrero.googlepages.com/software.

[19] S. Ratschan, "Efficient solving of quantified inequality constraints over the real numbers," *ACM Trans. on Computational Logic*, vol. 7, no. 4, pp. 723-748, 2006.

[20] M. Sainz, P. Herrero, J. Vehi, and J. Armengol, "Continuous minimax optimization using modal intervals," *Journal of Mathematical Analysis and Applications,* Elsevier Science, Oxford, UK, vol. 339, no. 1, pp. 18-30, 2006.

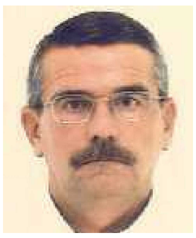
[21] A. Tarski, *A Decision Method for Elementary Algebra And Geometry*, Univ. of California Press, Berkeley, 1951.

Pau Herrero received the Ph.D. degree in Control Engineering from the University of Girona, Girona, Spain, in 2006 and from the University of Angers, Angers, France. In 2007, he was a postdoctoral Fellow, for one year, with the University of California, Santa Barbara, doing research on the development of an artificial pancreas. He is currently with the Hospital de la Santa Creu i Sant Pau, Barcelona, Spain, which belongs to the Network of Biomedical Research Centers in the field of Bioengineering, Biomaterials and Nanomedicine. His current research interests include the development and application of new technologies applied to the treatment of diabetes.



Luc Jaulin was born in Nevers, France in 1967. He received the Ph.D. degree in Automatic Control from the University of Orsay, France in 1993. He is currently a professor of Robotics at the ENSIETA engineering school in Brest, France, since 2004. He does his research on underwater robotics using interval methods and constraint propagation.

Josep Vehí was born in Girona, Spain, on January 9, 1964. He received the Ph.D. degree in Electrical Engineering from the Universitat de Girona, Girona, Spain in 1998. He was a teaching assistant, from 1987 to 1992, with the Technical University of Catalonia, Barcelona, Spain. Since 1992, he has been with the Universitat de Girona, where he was an assistant professor, from 1992 to 1999, and then an Associate Professor of electrical engineering in 1999. Since 2006, is the Director of the Research Institute on Computer Engineering and Automation, Universitat de Girona. His current research interests include interval methods for control, modeling and control of biomedical systems, robust and nonlinear control, fault detection and diagnostics for complex dynamic systems and intelligent methods for structural assessment.



Miguel A. Sainz was born in La Rioja, Spain, in 1942. He received the degree in Mathematics from the University of Zaragoza, Zaragoza, Spain, in 1964 and the Ph.D. degree from the University of Madrid, Madrid, Spain, in 1970. Since 1971, he has been a professor with the Polytechnic University of Catalonia, Barcelona, Spain, with the Autonomous University of Barcelona, Barcelona, Spain, and also with the University of Girona, Girona, Spain, where he is currently a member of the Research Institute on Computer Engineering and Automation, developing his research in modal interval analysis and its applications to problems of simulation and control. He presented several papers in international meetings (International Federation of Automatic Control World Congress, SAFEPROCESS) and published in scientific journals (Journal of Process Control and Reliable Computing).