Consistency techniques for the localization of a satellite

Luc Jaulin

Laboratoire d'Ingénierie des Systèmes Automatisés,
62 avenue Notre Dame du Lac 49 000 Angers, France `jaulin@univ-angers.fr`
`http://www.istia.univ-angers.fr/∼jaulin/`

Abstract. This paper recalls that the problem of estimating the state vector of a nonlinear dynamic system can be interpreted as a constraint satisfaction problem over continuous domains with a large number (several thousands) of variables and constraints. Consistency techniques are then shown to be particularly efficient to contract the domains for the variables involved. This is probably due to the large number of redundancies naturally involved in the constraints of the problem. The approach is illustrated on the estimation of the position of a satellite in orbit around the earth.

1 Introduction

The problem of estimating the position of a satellite in orbit around the earth is a problem of state estimation of a nonlinear dynamic system (see Section 3). The aim of this paper is to show that basic consistency techniques working on continuous domains [3], [4], [8], [7], [19] can be particularly efficient to solve this problem.

Most of the existing methods for state estimation compute their solutions on a linear approximation of the dynamic system ([13], [17], see also [15] in the context of localization). These methods are very efficient. The results obtained are solutions of the linear approximation and are expected to be also an approximation of the solution of the initial nonlinear problem. This is usually false when strong nonlinearities are involved. Some interval methods (but without propagation) have also been proposed in the context of state estimation (see, *e.g.*, [14]) but, to get an acceptable accuracy, a large number of bisections have to be performed, making the approach limited to low dimensional problems.

As linearization approaches, consistency techniques also compute an approximation of the solutions of the nonlinear problem. But their results are correct (*i.e.*, no solution is lost) and are not sensitive to strong nonlinearities or nondifferentiabilities in the dynamic system.

One of the main critic that can be done to consistency techniques is that they compute a superset of all solutions of the problem that may be huge even if the solution set is small. For state estimation, this pessimism is limited by presence of a large number of redundant constraints and the treatment of difficult nonlinear

state estimation problems can thus be considered. This can help to popularize consistency techniques. Note that some interesting results have already been obtained by consistency methods for the state estimation of a mobile robot in [12] or for the localization of a car in [2].

In Section 2 state estimation is shown to be a constraint satisfaction problem with many variables taking their values in the set of real numbers $\mathbb{R}$. In Section 3, the obtention of the state equations of the satellite is presented in details. A resolution of the localization problem is performed by the solver REALPAVER [6] in Section 4. Section 5 presents some possible improvements to increase the efficiency of consistency techniques.

2   State estimation is a constraint satisfaction problem

A huge class of dynamic systems can be described, after discretization of the corresponding differential equation, by the following equations:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k)) \\ \mathbf{y}(k) \qquad = \mathbf{g}(\mathbf{x}(k)). \end{cases} \tag{1}$$

See Figure 1 for an illustration.

- $k \in [0, k_{\max}]$ is an integer which corresponds to discrete time. If t is the continuous time, we have the relation $t = k\delta$ where δ is the sampling time.
- $\mathbf{u}(k) \in \mathbb{R}^{n_{\mathrm{u}}}$ is the input vector at time k. It can be chosen arbitrarily and is assumed to be known exactly.
- $\mathbf{y}(k) \in \mathbb{R}^{n_{\mathrm{y}}}$ is the output vector at time k. It is measured with a given precision depending on the sensors used, i.e., for each components $y_i(k)$ of $\mathbf{y}(k)$ a bounding interval $[y_i](k)$ is available.
- $\mathbf{x}(k) \in \mathbb{R}^{n_{\mathrm{x}}}$ is the state vector at time k. It can be assimilated to the memory of the system. It cannot be measured directly and nothing is known about it. State estimation aims at estimating its value, for all k.
- $\mathbf{w}(k) \in \mathbb{R}^{n_{\mathrm{w}}}$ is the perturbation vector at time k. It is unknown but bounded. Its components can usually be assumed to belong to a small interval $[w_i](k)$ containing zero. This vector takes into account unmodeled dynamics of the actual plant (relativist effects, . . .), unknown inputs (wind, . . .) or an error due to the discretization procedure.
- The first equation $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k))$ is called the *evolution equation* and describes the evolution of the system. Its form shows that the future of the system depends only of the state vector $\mathbf{x}(k)$ at the current time k and the present and future inputs $\mathbf{u}(k), \mathbf{u}(k+1), \ldots$ and perturbations $\mathbf{w}(k), \mathbf{w}(k+1), \ldots$ Thus, the vector $\mathbf{x}(k)$ summarizes everything that happened to the system in the past and that may influence the future behavior of the system.
- The second equation $\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k))$ is the *observation equation.* Each output $y_i(k)$ is a function the state vector at time k.

Remark 1. In practice, it is impossible to give a reliable measure of the error between the actual plant and the unperturbed model obtained by assuming that $\mathbf{w}(k) = 0$ in (1). Thus the domains $[w_i](k)$ chosen for the perturbation variables are not guaranteed to contain the true values for $w_i(k)$. The results obtained by a reliable method may thus be unreliable. When no solution exists for our state estimation problem, we are certain that too optimistic bounds have been chosen for $w_i(k)$. The associated domains should then be enlarged (see Section 5).

$$\mathbf{w}(k)$$

$$
\boxed{
\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k), \mathbf{w}(k), \mathbf{u}(k)) \\
\mathbf{y}(k) &= \mathbf{g}(\mathbf{x}(k))
\end{aligned}
}
$$

$\mathbf{u}(k)$ →   → $\mathbf{y}(k)$
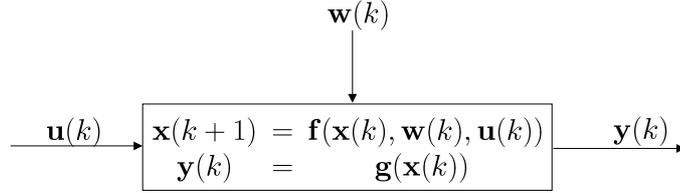
Fig. 1. Illustration of a discrete-time dynamic system

Our problem can be represented (see [12]) by a *constraint satisfaction problem* (CSP) where the set of $(k_{\max} + 2)\, n_x + (k_{\max} + 1)\, (n_w + n_y)$ variables is

$$
\begin{aligned}
\mathcal{V} = \{ & x_1(0), \ldots, x_{n_x}(0), x_1(1), \ldots, x_{n_x}(1), \ldots, \\
& \ldots, x_1(k_{\max} + 1), \ldots, x_{n_x}(k_{\max} + 1), \\
& w_1(0), \ldots, w_{n_w}(0), w_1(1), \ldots, w_{n_w}(1), \ldots, \\
& \ldots, w_1(k_{\max}), \ldots, w_{n_w}(k_{\max}), \\
& y_1(0), \ldots, y_{n_y}(0), y_1(1), \ldots, y_{n_y}(1), \ldots, \\
& \ldots, y_1(k_{\max}), \ldots, y_{n_y}(k_{\max}) \},
\end{aligned}
\tag{2}
$$

the set of domains is

$$
\begin{aligned}
\mathcal{D} = \{ & [x_1](0), \ldots, [x_{n_x}](0), [x_1](1), \ldots, [x_{n_x}](1), \ldots, \\
& \ldots, [x_1](k_{\max} + 1), \ldots, [x_{n_x}](k_{\max} + 1), \\
& [w_1](0), \ldots, [w_{n_w}](0), [w_1](1), \ldots, [w_{n_w}](1), \ldots, \\
& \ldots, [w_1](k_{\max}), \ldots, [w_{n_w}](k_{\max}), \\
& [y_1](0), \ldots, [y_{n_y}](0), [y_1](1), \ldots, [y_{n_y}](1), \ldots, \\
& \ldots, [y_1](k_{\max}), \ldots, [y_{n_y}](k_{\max}) \},
\end{aligned}
\tag{3}
$$

and the set of $(k_{\max} + 1)\, (n_x + n_y)$ constraints is

$$
\begin{aligned}
\mathcal{C} = \{ & x_1(1) = f_1\left(x_1(0), \ldots, x_{n_x}(0), w_1(0), \ldots, w_{n_w}(0), u_1(0), \ldots, u_{n_u}(0)\right), \\
& \ldots, \\
& x_{n_x}(1) = f_{n_x}\left(x_1(0), \ldots, x_{n_x}(0), w_1(0), \ldots, w_{n_w}(0), u_1(0), \ldots, u_{n_u}(0)\right), \\
& y_1(0) = g_1\left(x_1(0), \ldots, x_{n_x}(0)\right), \ldots, \\
& y_{n_y}(0) = g_{n_y}\left(x_1(0), \ldots, x_{n_x}(0)\right), \\
& \vdots \qquad\qquad \},
\end{aligned}
$$

where $f_1, \ldots, f_{n_x}$ and $g_1, \ldots, g_{n_y}$ are the coordinate functions associated with $\mathbf{f}$ and $\mathbf{g}$. Note that since the $u_i(k)$ are assumed to be exactly known, they have not been considered as a variable of the CSP. In our formulation, it as been assumed that for all k, domains $[x_i](k), [w_i](k)$ and $[y_i](k)$ containing the variables $x_i(k), w_i(k)$ and $y_i(k)$ are available. This is true for the measured outputs $y_i(k)$, this is also true for the bounded perturbations $w_i(k)$, but nothing is known about the values of the values of the state variable $x_i(k)$. Therefore, the corresponding domains should be taken as $[x_i](k) =]-\infty, \infty[$.

3   Formulation of satellite localization's problem

As an application, consider the satellite, with coordinates (x_1, x_2) in orbit around the earth as illustrated on Figure 2.
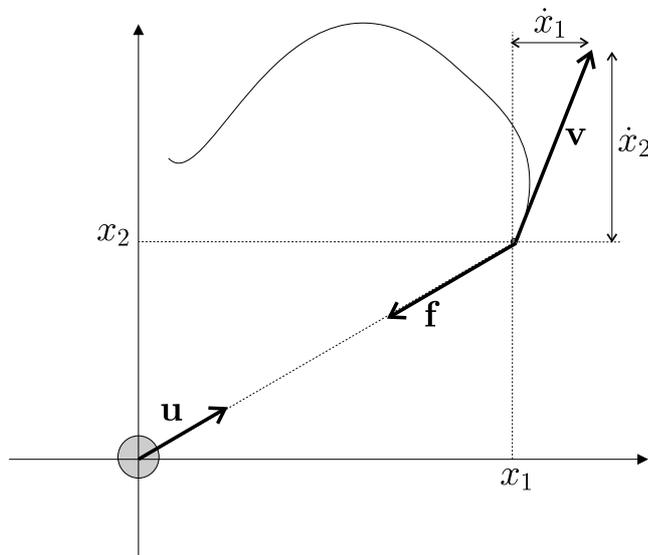
Fig. 2. Earth at coordinate $(0,0)$ and the satellite at coordinate (x_1, x_2)

Newton's universal law of gravitation states that the earth exerts a gravitational force $\mathbf{f}$ of attraction on the satellite. The direction of $\mathbf{f}$ is along the unit vector $\mathbf{u}$ directed by the line joining the two objects. The magnitude of the force is proportional to the product of the gravitational masses of the objects, and inversely proportional to the square of the distance between them. We have

$$\mathbf{f} = -G \frac{Mm}{x_1^2 + x_2^2} \mathbf{u}, \tag{4}$$

where M is the mass of the earth, m is the mass of the satellite and $G = 6.67x10^{-11}Nm^2kg^{-2}$ is the Newton's constant. Since

$$\mathbf{u} = \frac{1}{\sqrt{x_1^2 + x_2^2}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$ (5)

we have

$$\mathbf{f} = -G\frac{Mm}{\sqrt{x_1^2 + x_2^2}^3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$ (6)

Now, from the Newton's second law, we have

$$\mathbf{f} = m\frac{d\mathbf{v}}{dt} = m\begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix}.$$ (7)

where $\mathbf{v}$ is the velocity of the satellite. Thus

$$\begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix} = -G\frac{M}{\sqrt{x_1^2 + x_2^2}^3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$ (8)

Set $\alpha = GM$, then

$$\begin{cases} \ddot{x}_1 = -\frac{\alpha x_1}{\sqrt{x_1^2+x_2^2}^3} \\ \ddot{x}_2 = -\frac{\alpha x_2}{\sqrt{x_1^2+x_2^2}^3} \end{cases}$$ (9)

For simplicity, we shall assume that $\alpha = 1$. To obtain a first order equation (to get state equations), we have to introduce two other variables x_3 and x_4 which satisfy

$$\begin{cases} \dot{x}_1 = x_3 \\ \dot{x}_2 = x_4 \end{cases}$$ (10)

Then we have the following continuous state equations:

$$\begin{cases} \dot{x}_1 = x_3 \\ \dot{x}_2 = x_4 \\ \dot{x}_3 = -\frac{x_2}{\sqrt{x_1^2+x_2^2}^3} \\ \dot{x}_4 = -\frac{x_1}{\sqrt{x_1^2+x_2^2}^3} \end{cases}$$ (11)

In practice, we are not able to measure the state directly. Instead, we shall assume that we are able to measure three quantities:

- y_1 : the angle between the direction satellite and a fixed direction (here, we took the tangent y_1 of the angle to get polynomial relations);
- y_2 : the radius speed of the satellite (which is equal to zero if the distance between the satellite and the earth remains constant). This can be done by the Doppler shift method which applies to electromagnetic waves in all portions of the spectrum. Radiation is blueshifted when the satellite moves away, and is redshifted when it comes near the earth;

– y_3 : the distance between the satellite and the earth.

The observation equations can thus be described by:

$$\begin{cases} y_1 = x_2/x_1 \\ y_2 = x_1 x_3 + x_2 x_4 \\ y_3 = x_1^2 + x_2^2 \end{cases} \tag{12}$$

In order to get discrete-time state equation, a discretization of the continuous state equation (11) should be performed. Using a first order approximation:

$$\mathbf{x}(t + \delta) \simeq \mathbf{x}(t) + \delta \mathbf{f}\left(x\left(t\right)\right), \tag{13}$$

where the sampling time is taken as $\delta = 0.5\,\mathrm{sec}$, from (11) and (12), we get:

$$\begin{cases} x_1(k+1) = x_1(k) + 0.5\ x_3(k) + w_1(k) \\ x_2(k+1) = x_2(k) + 0.5\ x_4(k) + w_2(k) \\ x_3(k+1) = x_3(k) - 0.5\ \dfrac{x_1(k)}{\sqrt{x_1^2(k)+x_2^2(k)}^3} + w_3(k) \\ x_4(k+1) = x_4(k) - 0.5\ \dfrac{x_2(k)}{\sqrt{x_1^2(k)+x_2^2(k)}^3} + w_4(k) \\ y_1(k) \qquad = x_2(k)/x_1(k) \\ y_2(k) \qquad = x_1(k)x_3(k) + x_2(k)x_4(k) \\ y_3(k) \qquad = x_1^2(k) + x_2^2(k) \end{cases} \tag{14}$$

Here, we have added perturbation variables w_1, w_2, w_3 and w_4 to take into account errors due to the discretization procedure and other unmodeled dynamics (presence of other planets, tide effects, ...). Note that it is impossible in practice to find rigorous bounds for w_i. For our example, we shall take $[w_i](k) = [-0.01, 0.01]$. For the output, the sensors are usually capable of giving rigorous intervals for the actual output $y_i(k)$. Here, we shall assume that the domains $[y_i(k)]$ for $y_i(k)$ is given by $[\check{y}_i(k) - 0.1, \check{y}_i(k) + 0.1]$, where $\check{y}_i(k)$ is the measure (*i.e.*, an approximation) of $y_i(k)$.

Again, the formulation of our system is a constraint satisfaction problem. For $k_{\max} = 200$, the set of 2215 variables is

$$\begin{aligned} \mathcal{V} = \{ &x_1(0), x_2(0), x_3(0), x_4(0), \ldots, x_1(201), x_2(201), x_3(201), x_4(201), \\ &w_1(0), w_2(0), w_3(0), w_4(0), \ldots, w_1(200), w_2(200), w_3(200), w_4(200), \\ &y_1(0), y_2(0), y_3(0), \ldots, y_1(200), y_2(200), y_3(200)\}, \end{aligned} \tag{15}$$

the set of domains is

$$\begin{aligned} \mathcal{D} = \{ &]-\infty, \infty[,]-\infty, \infty[, \ldots,]-\infty, \infty[, \\ &[-0.01, 0.01], [-0.01, 0.01], \ldots, [-0.01, 0.01], \\ &[\check{y}_1(0) - 0.1, \check{y}_1(0) + 0.1], \ldots, [\check{y}_3(200) - 0.1, \check{y}_3(200) + 0.1]\}. \end{aligned} \tag{16}$$

The set of 1407 constraints of the CSP is given by the state equations given in formula (14).

Remark 2. The measurements $\check{y}_i(k)$ have been generated by simulation (with (14)) with the following initial conditions

$$\mathbf{x}(0) = (4,\ 0.5,\ -0.2,\ 0.5)^{\mathrm{T}}.$$

Note that other initial conditions could have lead to the same measures. This is the case for the initial vector

$$\mathbf{x}'(0) = (-4,\ -0.5,\ 0.2,\ -0.5)^{\mathrm{T}}.$$

The fact that different initial solutions can generate the same measurements (even in a noise-free context) is known as an identifiability problem [20]. Therefore, even without noise we will never be able to compute the initial conditions in a unique way. To avoid this identifiability problem we shall restrict the domains for $x_1(0), x_2(0), x_3(0)$ and $x_4(0)$ to $[0, \infty[, [0, \infty[,]-\infty, 0]$ and $[0, \infty[$, respectively.

Remark 3. Although the CSP described above contains 1407 constraints for 2215 variables, it can be considered as redundant. This is due to the fact that many of these variables (here exactly 1407 : the measurements $y_i(k)$ and the perturbation $w_i(k)$) have small domains and can thus be considered as approximately known.

4 Resolution

The contraction of the domains have been performed by the solver REALPAVER by using a basic local consistency method. Some lines of the program accepted by REALPAVER are given by:

```
Bisection none;
x1_0 in [0,+oo[,
x2_0 in [0,+oo[,
x3_0 in [-oo,0[,
x4_0 in [0,+oo[,
w1_0 in [-0.01,0.01],
w2_0 in [-0.01,0.01],
w3_0 in [-0.01,0.01],
w4_0 in [-0.01,0.01],
y1_0 in [0.107329,0.307329],
y2_0 in [-0.71049,-0.51049],
y3_0 in [16.117,16.317],
x1_1 in ]-oo,+oo[,
x2_1 in ]-oo,+oo[,
x3_1 in ]-oo,+oo[,
x4_1 in ]-oo,+oo[,
...
w1_200 in [-0.01,0.01],
w2_200 in [-0.01,0.01],
w3_200 in [-0.01,0.01],
```

```
w4_200 in [-0.01,0.01],
y1_200 in [-2.42001,-2.22001],
y2_200 in [-0.925105,-0.725105],
y3_200 in [136.612,136.812],
x1_201 in ]-oo,+oo[,
x2_201 in ]-oo,+oo[,
x3_201 in ]-oo,+oo[,
x4_201 in ]-oo,+oo[;

Constraints

x1_1=x1_0+0.5*x3_0+ w1_0,
x2_1=x2_0+0.5*x4_0+ w2_0,
x3_1=x3_0-0.5*x1_0/(sqrt(x1_0^2+x2_0^2))^3+w3_0,
x4_1=x4_0-0.5*x2_0/(sqrt(x1_0^2+x2_0^2))^3+w4_0,
y1_0=x2_0/x1_0,
y2_0=x1_0*x3_0+x2_0*x4_0,
y3_0=x1_0^2+x2_0^2,
...
x1_201=x1_200+0.5*x3_200+ w1_200,
x2_201=x2_200+0.5*x4_200+ w2_200,
x3_201=x3_200-0.5*x1_200/(sqrt(x1_200^2+x2_200^2))^3+w3_200,
x4_201=x4_200-0.5*x2_200/(sqrt(x1_200^2+x2_200^2))^3+w4_200,
y1_200=x2_200/x1_200,
y2_200=x1_200*x3_200+x2_200*x4_200,
y3_200=x1_200^2+x2_200^2;
```

The whole program can be downloaded at *http://www.istia.univ-angers.fr/˜jaulin/satellite.txt*. On a Pentium 300, REALPAVER returns the following lines

```
  x1_0 in [3.8914121266466 , 4.017994154622222]
  x2_0 in [0.4176613721408529 , 0.9567337119326976]
  x3_0 in [-0.4113978603389778 , -0.1470621703041645]
  x4_0 in [0.2009142731412085 , 0.9312204034312483]
  y1_0 in [0.107329 , 0.2536019303090482]
  y2_0 in [-0.7104900000000001 , -0.51049]
  y3_0 in [16.117 , 16.317]
...
  x1_200  in [4.385557786959884 , 4.885898743627584]
  x2_200  in [-10.84674406984067 , -10.61309370002079]
  x3_200  in [-0.1485890863687774 , +0.541771027979396]
  x4_200  in [-0.0002353318428811776 , +0.3258643220373622]
  y1_200  in [-2.42001 , -2.22001]
  y2_200  in [-0.9251050000000001 , -0.725105]
  y3_200  in [136.612 , 136.812]
...
END OF SOLVING
```

```
Property:      reliable process (no solution is lost)
Elapsed time: 810 ms
```

Figure 3 represents the actual values for the positions $(x_1(k), x_2(k))$ for the satellite and for $k \in [0, 200]$ that have been obtained by a simulation.
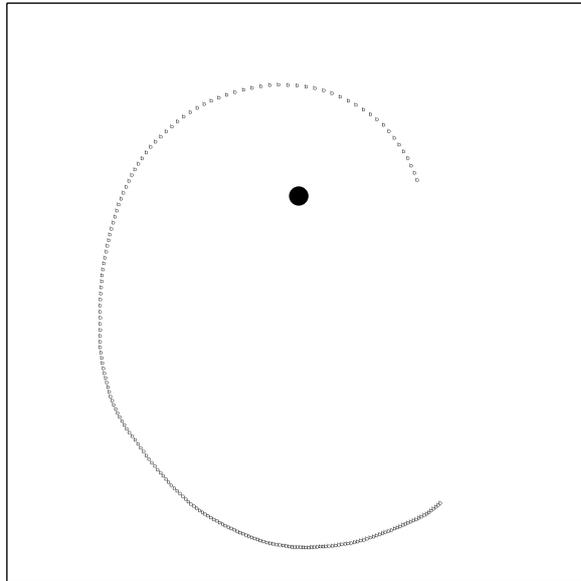
Fig. 3. Actual position of the satellite for k varying from 0 to 200

Figure 4 represents the boxes $[x_1](k) \times [x_2](k)$. This figure is consistent with the fact that for all k, the actual position vector $(x_1(k), x_2(k))$ obtained by the simulation is an element of the box $[x_1](k) \times [x_2](k)$. For some k, the approximation obtained is too rough. This is probably due to the fact that in the associated regions for the satellite, the measures are not informative enough and that the local consistency used is not strong enough to get an accurate enclosure for the state vector.

Figure 5 (left) represents the computing time (t_c) of REALPAVER for different $k_{\max}$. The right part represents $t_c/k_{\max}$ with respect to $k_{\max}$. These experimental results suggest that the complexity of the local consistency algorithm used by REALPAVER could be quadratic with respect the number of variables. Note that when $k_{\max} = 800$, the CSP treated by REALPAVER is huge since it contains 8815 variables for 5607 constraints. To my knowledge, such a huge CSP related to an application has never been treated before.

Remark 4. The constraint graph has here a rather uncommon structure (studied in [11]) which is not taken into account by REALPAVER. A more efficient
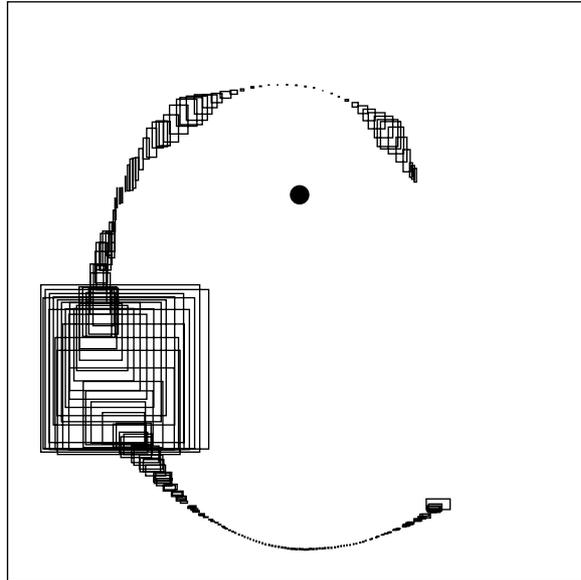
Fig. 4. Set of boxes generated by REALPAVER containing the actual positions for the satellite for different k in $[0, 200]$
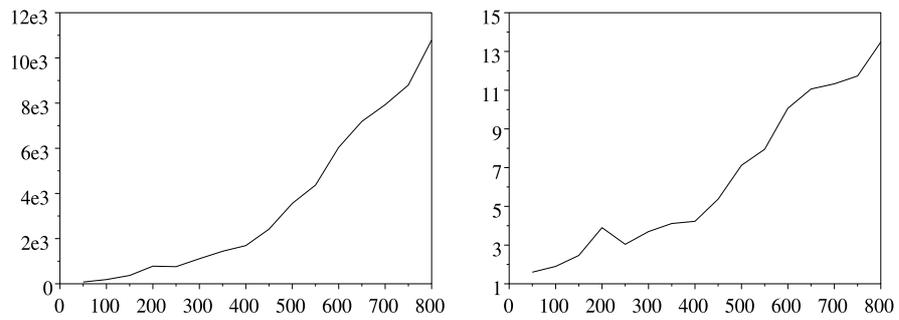
Fig. 5. Left: computing time t_c (in ms) with respect to $k_{\max}$; Right, $t_c/k_{\max}$ with respect to $k_{\max}$

approach would be to alternate a forward and a backward propagation until a fixed point is reached. This approach has been presented in [10].

5   Possible improvements

5.1   Using linear consistencies

The contraction method used for the localization of the satellite is based on hull consistency. Now, hull consistency is local (it considers the constraints independently) and pessimism exists when a variable occurs more than once in a constraint (due to the dependency problem). This pessimism can be reduced by using box-consistency [1]. More global (i.e., taking the dependency between two or more constraints into account) consistency approaches can also be considered [16], [18], but most of them are very sensitive to the number of variables which is very high for state estimation problems.

Now, for most state estimation problems, it is always possible to find large subnetworks of the associated problem which involve only linear constraints. The domains for the variables of the linear subnetworks can then be contracted by linear techniques and propagated back to the original CSP. This is illustrated by the following example

Example 1. Consider the example of a polynomial CSP, inspired by our satellite localization problem, with 7 variables and 3 constraints given by

$$\mathcal{C} : \begin{cases} x'_1 = x_1 + x_3 \\ x'_2 = x_4 \\ x'_3 = x_3 - \frac{x_1}{\sqrt{x_1^2 + x_2^2}^3} \end{cases} \tag{17}$$

This set of constraints can be decomposed into the 8 following primitive constraints:

$$\mathcal{C}_p : \begin{cases} x'_1 = x_1 + x_3 \\ x'_2 = x_4 \\ x'_3 = x_3 - z_5 \\ z_1 = x_1^2 \\ z_2 = x_2^2 \\ z_3 = z_1 + z_2 \\ z_4 = z_3^{-3/2} \\ z_5 = x_1 z_4 \end{cases} \tag{18}$$

Assume that all domains of the nonlinear variables of $\mathcal{C}_p$ are included in $]0, \infty[$. If this is not the case, the constraint involving a variable that may be negative should be removed. Then $\mathcal{C}_p$ can be decomposed as follows:

$$\mathcal{C}_1 : \begin{cases} x'_1 = x_1 + x_3 \\ x'_2 = x_4 \\ x'_3 = x_3 - z_5 \\ z_3 = z_1 + z_2 \end{cases} \quad \mathcal{C}_2 : \begin{cases} \hat{z}_1 = \log(z_1) \\ \hat{z}_2 = \log(z_2) \\ \hat{z}_4 = \log(z_3) \\ \hat{z}_5 = \log(z_4) \\ \hat{x}_1 = \log(x_1) \\ \hat{x}_2 = \log(x_2) \end{cases} \quad \mathcal{C}_3 : \begin{cases} \hat{z}_1 = 2\hat{x}_1 \\ \hat{z}_2 = 2\hat{x}_2 \\ \hat{z}_4 = -\frac{3}{2}\hat{z}_3 \\ \hat{z}_5 = \hat{x}_1 + \hat{z}_4 \end{cases} \tag{19}$$

A linear solver can thus be used to contract optimally $\mathcal{C}_1$ and $\mathcal{C}_3$. A local consistency technique is here optimal for $\mathcal{C}_2$. The resulting contraction can then be used to contract the original CSP (17).

5.2   Tuning the initial size for the domains

In practice, an efficient contraction method is needed but global consistency is not always desirable. Indeed, our model is only an approximation of the reality and ignores many unknown effects. As a result, the solution set of the CSP is often empty and global consistency would result in a loss of any information we had about the position of the satellite. A possible way to keep a satisfactory result is to use a parameter β to tune the size of the perturbation and output intervals. This parameter is decreased until the envelope generated by our favorite contraction method is accurate enough. For our satellite problem, the following algorithm can be chosen.

1. $\beta = \beta_0$, $\ell = 1$;
2. for $i \in \{1, 2, 3, 4\}, k \in \{0, \ldots, 200\}$, $[w_i(k)] = [-0.01\beta, 0.01\beta]$;
3. for $i \in \{1, 2, 3\}, k \in \{0, \ldots, 200\}$, $[y_i(k)] = [\check{y}_i(k) - 0.1\beta, \check{y}_i(k) + 0.1\beta]$;
4. $\mathcal{S}(\ell) =$ approximation of the solution set of the CSP obtained by the available consistency method;
5. if $\mathcal{S}(\ell) \neq \emptyset$, $\ell = \ell + 1$; decrease β, go to 2;
6. else return $\mathcal{S}(\ell - 1)$; END;

In this algorithm, β_0 is large enough to make to CSP satisfiable at the first step. This algorithm

– never returns the empty set. This is important for the engineer, who often needs a punctual estimation of for state vectors in order to take his decision,
– can tackle nonlinear constraints much more easily than most existing methods which are not based on consistency algorithm,
– can produce a result with an accurate approximation in a short time, depending on the consistency used.

Of course, the algorithm is not anymore correct but the correctness is not as fundamental for engineer: since many unknown effects (relativity, presence of other planets, shape of the earth, ...) are always neglected in the equations, the model is never correct. However, I believe that the above algorithm may often produce *more correct* results than most existing nonlinear approaches which are generally based on the linearization of the equations of the dynamic system.

6   Conclusion

In this paper, consistency techniques have been used to deal with the difficult problem of nonlinear state estimation which has been recalled to be a constraint

satisfaction problem with continuous variables. The large number of measurements that can generally be performed on the dynamic system introduces many redundancies which benefit to constraint propagation methods. Moreover, since in practice bounds of some variables can be violated and since many other unknown effects of the dynamic system are neglected, the solution set of the CSP to be solved is generally empty. The pessimism of consistency methods may thus counterbalance the fact that no solution exists. An elementary algorithm to tune the balance parameter β has been proposed to give satisfaction to engineers who need a reasonably small (but nonempty) envelope for the state vectors to the detriment of the correctness of the results.

The approach presented here has supposed that the dynamic system was described by a discrete-time state equation. Now, in practice, the system is described by nonlinear differential equations. The discretization procedure adds some errors that could be avoided. An efficient approach working directly on the differential equation could be obtained by combining the approach presented in [5] with that in [9].

References

1. Benhamou, F., Goualard, F., Granvilliers, L., Puget, J.F.: Revising Hull and Box Consistency. Proceedings of the International Conference on Logic Programming, Las Cruces, NM, (1999) 230–244
2. Bouron, R.: Méthodes ensemblistes pour le diagnostic, l'estimation d'état et la fusion de données temporelles. Université de Compiègne, PhD dissertation, (2002)
3. Cleary, J.G.: Future Computing Systems. Logical arithmetic, **2**(2), (1987) 125–149
4. E. Davis, E.: Constraint propagation with interval labels. Artificial Intelligence, **32**(3) (1987) 281–331
5. Deville, Y., Janssen, M., Van Hentenryck, P.: Consistency techniques in ordinary differential equations. Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming. Lecture Notes in Computer Science, Springer Verlag, Berlin Heidelberg New York (1998)
6. Granvilliers, L.: RealPaver, available at , `http://www.sciences.univ-nantes.fr /info/perso/permanents/granvil/realpaver/`. IRIN, University of Nantes (2002)
7. van Hentenryck, P., Deville, Y., Michel, L.: Numerica: A Modeling Language for Global Optimization. MIT Press, Boston, MA, (1997)
8. Hyvönen, E.: Constraint reasoning based on interval arithmetic; The tolerance propagation approach. Artificial Intelligence, **58**(1-3), (1992) 71-112
9. Jaulin, L.: Nonlinear bounded-error state estimation of continuous-time systems. Automatica. **38** (2002) 1079–1082
10. Jaulin, L., Braems, I., Kieffer, M., Walter, E.: Nonlinear State Estimation Using Forward-Backward Propagation of Intervals. W. Kraemer and J.W. Gudenberg (Eds), Scientific Computing, Validated Numerics, Interval Methods, Proceedings of SCAN 2000, Kluwer Academic Publishers (2001) 191-204
11. Jaulin, L., Kieffer, M., Braems, I., Walter E.: Guaranteed Nonlinear Estimation Using Constraint Propagation on Sets. International Journal of Control, **74** (18), (2001) 1772–1782
12. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics, Springer-Verlag, London, (2001)

13. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the AMSE, Part D, Journal of Basic Engineering, **82** (1960) 35–45

14. Kieffer, M., Jaulin, L., Walter, E., Meizel, D.: Robust Autonomous Robot Localization Using Interval Analysis. Reliable Computing, **6**(3), (2000) 337–362

15. Leonard, J.J., Durrant-Whyte, H.F.: Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation, **7**(3), (1991) 376–382

16. Lhomme, O.: Consistency Techniques for Numeric CSPs. Proceedings of the International Joint Conference on Artificial Intelligence, Chambéry, France, (1993) 232–238

17. Ljung, L.: Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems. IEEE Transactions on Automatic Systems, **24**, (1979) 36–50

18. Sam-Haroud, D.J., Faltings, B.: Consistency Techniques for Continuous Constraints. Constraints, **1**(1-2), (1996) 85–118

19. VanEmden, M.: Algorithmic Power from Declarative Use of Redundant Constraints. Constraints, **4**(4), (1999) 363–381

20. Walter, E.,Identifiability of state space models, Springer-Verlag, Berlin, Germany, (1982)