

Exact bounded-error continuous-time linear state estimator

Simon Rohou, Luc Jaulin

▶ **To cite this version:**

Simon Rohou, Luc Jaulin. Exact bounded-error continuous-time linear state estimator. Systems and Control Letters, 2021, 153, pp.104951. 10.1016/j.sysconle.2021.104951 . hal-03242585

HAL Id: hal-03242585

https://ensta-bretagne.hal.science/hal-03242585

Submitted on 31 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exact bounded-error continuous-time linear state estimator

Simon Rohou[a], Luc Jaulin[a]

[a]*ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France*

Abstract

This paper proposes an interval-based method for estimating the state of a linear continuous-time dynamical system. In this work, we assume that the measurements are provided at discrete times and that all errors are bounded. Interval analysis is used to propagate the interval uncertainties continuously over time. The resulting method is guaranteed to never lose any feasible solution and provides an optimal polygonal enclosure of the state trajectory. A reproducible example illustrates the principle of the method.

Keywords: linear system, state-space equations, interval analysis, tubes, time uncertainties, state estimator.

1. Introduction

This paper proposes a new state estimator for linear time-invariant systems described by state equations. The measurements are collected at discrete times and all errors are assumed to be intervals. The states are known to belong to some prior sets, possibly infinite if nothing is known *a priori*. Using interval analysis, we show that it is possible to obtain an accurate and guaranteed enclosure of the state vectors that is consistent with the state equation, the observations and the related errors. This state estimator is *exact* since it does not introduce pessimism and does not lose any consistent state.

The state estimation of linear systems is usually treated by a Kalman filter [1, 2]. Since several decades, the Kalman approach has been adapted to consider bounded-error systems [3, 4, 5]. This set-membership approach is interesting when we have few measurements [6] or when the noise is more properly described by intervals rather than a probabilistic distribution. For the resolution, different types of wrappers have been used to enclose the feasible states: ellipsoids [7, 8], intervals [9, 10], zonotopes [11, 12, 13], parallelotopes [14], or polytopes [15, 16]. All these contributions consider a discrete time evolution.

The approach proposed in this paper is an extension of the method introduced in [16], which is dedicated to discrete time systems. This method has been shown to be exact and can be written in a recursive form. Now, the main difficulty is to take into account the continuous time, the discrete measurements and the accuracy of the enclosure, in one algorithm.

Dealing with continuous time has already been considered by several authors such as [17]. Our main contribution is to propose a state estimator based on interval analysis that computes exactly the set of all feasible trajectories. To our knowledge, none of the existing state estimators is able to compute this exact set in the case of continuous linear systems. They either require the system to be discrete in time (see *e.g.*, [16]) or they introduce some conservatism [18]. The overestimation of linear continuous-time state estimator is mainly due to the fact that necessary conditions [19] are used, such as positivity, see *e.g.* [20], or wrapping effects.

This paper is organized as follows. Section 2 introduces the exact sequence defining the set of all states consistent with the errors and the differential equation associated with the evolution. Section 3 specifies the exact sequence in the case where the system is linear and time-invariant. It also proposes a polygonal algorithm for computing an outer approximation of the exact state trajectory. Finally, an example related to the localization of a mobile robot is treated in Section 4, before the conclusion of the paper.

2. Exact sequence

2.1. Tubes

Denote by $\mathcal{F}\left(\mathbb{R} \mapsto \mathbb{R}^n\right)$ the set of trajectories with values in $\mathbb{R}^n$. A tube [21] is a set of trajectories $\mathbb{X}(\cdot) : \mathbb{R} \mapsto \mathcal{P}(\mathbb{R}^n)$ which satisfies

$$\mathbf{x}(\cdot) \in \mathbb{X}(\cdot) \Leftrightarrow \forall t, \mathbf{x}(t) \in \mathbb{X}(t).$$

This definition shows that any value $\mathbf{x}(t)$ is enclosed in a set. Nevertheless, there is no additional information related to temporal correlations or derivatives with respect to t. For instance, the set of trajectories $\mathbb{A}(\cdot) : t \mapsto [-1, 1]$ is a tube, and the trajectory $a(t) = \sin(t)$ is enclosed in $\mathbb{A}(\cdot)$ since it satisfies $a(t) \in \mathbb{A}(t), \forall t$. However, the set $\mathbb{B}(\cdot)$

Email addresses: simon.rohou@ensta-bretagne.fr (Simon Rohou), luc.jaulin@gmail.com (Luc Jaulin)

of all trajectories $b(\cdot)$, such that $b(t) \in [-1,1]$ and $\dot{b}(t) \geqslant 0$, is not a tube.

Now, let us define the inclusion between two tubes as

$$\mathbb{X}(\cdot) \subset \mathbb{Y}(\cdot) \Leftrightarrow \forall t, \mathbb{X}(t) \subset \mathbb{Y}(t).$$

The set of tubes equipped with the inclusion is a lattice [22]. As a consequence, if we define a set of trajectories $\mathbb{B}(\cdot)$, there exists a unique smallest tube that encloses $\mathbb{B}(\cdot)$. For instance, the smallest tube enclosing all trajectories $b(\cdot)$ defined in $t \mapsto [-1,1]$ such that $\dot{b}(t) \geqslant 0$ is the tube $\mathbb{A}(\cdot) : t \mapsto [-1,1]$.

These definitions also apply when t is restricted to be inside an interval $[0,\bar{t}]$, representing a temporal domain.

2.2. The input $\mathbf{u}(\cdot)$ is known

Consider the linear time-invariant dynamical system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \tag{1}$$

Because the system is linear, an analytical expression for the flow is given by

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}) = e^{\mathbf{A}(t_2-t_1)}\mathbf{x} + \int_{t_1}^{t_2} e^{\mathbf{A}(t_2-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau. \tag{2}$$

Assume that for $t \in [0,\bar{t}]$, the state trajectory $\mathbf{x}(\cdot)$ is known to be inside the prior tube $\check{\mathbb{X}}(\cdot)$. We want to compute recursively the smallest tube $\mathbb{X}(\cdot)$ for $\mathbf{x}(\cdot)$ consistent with both the prior tube $\check{\mathbb{X}}(\cdot)$ and the state equation. This sequence can be interpreted as an extension to continuous time systems of the state estimator proposed in [23] or [24].

For a given initial vector $\mathbf{x}_1$ defined at t_1, the state at time t_2 is expressed by

$$\mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1),$$

where $\mathbf{\Phi}_{t_1,t_2}$ is the flow of the dynamical system according to its input $\mathbf{u}(\cdot)$. The flow can be extended to sets [25] as follows:

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) = \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_1 \in \mathbb{X}_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1)\right\}. \tag{3}$$

The *dot* notation $(\cdot)$ is used in this paper for representing the independent variable. Also, note that for ease of reading, the notation $\mathbf{u}(\cdot)$ is used in the above equation instead of $\mathbf{u}_{[t_1,t_2]}(\cdot)$, as there is no dependency with the values of $\mathbf{u}(\cdot)$ outside the interval $[t_1,t_2]$.

The following two corollaries will be used thereafter for the proof of Theorem 3.

Corollary 1. *Given a set $\mathbb{X}_1$, three time instants t_1, t_2, t_3, and an input $\mathbf{u}(\cdot)$, we have the Chasles property:*

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_3}(\mathbb{X}_1) \quad = \quad \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_2,t_3} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1)\ . \tag{4}$$

Proof. From Equation (3), we have:

$$\begin{aligned}
&\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_2,t_3} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) \\
&\quad = \left\{\mathbf{x}_3 \,|\, \exists \mathbf{x}_2 \in \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1), \mathbf{x}_3 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_2,t_3}(\mathbf{x}_2)\right\} \\
&\quad = \left\{\mathbf{x}_3 \,|\, \exists \mathbf{x}_1 \in \mathbb{X}_1, \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1), \mathbf{x}_3 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_2,t_3}(\mathbf{x}_2)\right\} \\
&\quad = \left\{\mathbf{x}_3 \,|\, \exists \mathbf{x}_1 \in \mathbb{X}_1, \mathbf{x}_3 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_2,t_3} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1)\right\} \\
&\quad = \left\{\mathbf{x}_3 \,|\, \exists \mathbf{x}_1 \in \mathbb{X}_1, \mathbf{x}_3 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_3}(\mathbf{x}_1)\right\} \\
&\quad = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_3}(\mathbb{X}_1)
\end{aligned}$$

□

Corollary 2. *Given two sets $\mathbb{X}^a_1, \mathbb{X}^b_1$, two time instants t_1, t_2, and an input $\mathbf{u}(\cdot)$, we have the automorphism property:*

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^a_1 \cap \mathbb{X}^b_1) \quad = \quad \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^a_1) \cap \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^b_1). \tag{5}$$

Proof. From Equation (3), we have:

$$\begin{aligned}
&\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^a_1 \cap \mathbb{X}^b_1) \\
&\quad = \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_1 \in \mathbb{X}^a_1 \cap \mathbb{X}^b_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1)\right\} \\
&\quad = \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_{1a} \in \mathbb{X}^a_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_{1a})\right. \\
&\qquad \left. \wedge\ \exists \mathbf{x}_{1b} \in \mathbb{X}^b_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_{1b})\right\} \\
&\quad = \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_{1a} \in \mathbb{X}^a_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_{1a})\right\} \\
&\qquad \cap \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_{1b} \in \mathbb{X}^b_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_{1b})\right\} \\
&\quad = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^a_1) \cap \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^b_1)
\end{aligned}$$

□

We define the *posterior state tube* as the smallest tube $\hat{\mathbb{X}}(\cdot)$ for $\mathbf{x}(\cdot)$ consistent with the prior tube $\check{\mathbb{X}}(\cdot)$, the input $\mathbf{u}(\cdot)$ and the state equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$. We have

$$\hat{\mathbb{X}}_t = \bigcap_{\tau \in [0,\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau), \tag{6}$$

as illustrated by Figure 1.

We now define the set

$$\overrightarrow{\mathbb{X}}_t = \bigcap_{\tau \in [0,t]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \tag{7}$$

that corresponds to the set of all $\mathbf{x}(t)$ consistent with the past (before t) and

$$\overleftarrow{\mathbb{X}}_t = \bigcap_{\tau \in [t,\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \tag{8}$$

the set of all $\mathbf{x}(t)$ consistent with the future (after t). We have

$$\hat{\mathbb{X}}_t = \overrightarrow{\mathbb{X}}_t \cap \overleftarrow{\mathbb{X}}_t. \tag{9}$$

The following theorem will allow us to implement the *exact sequence* to compute the tube $\hat{\mathbb{X}}(\cdot)$.

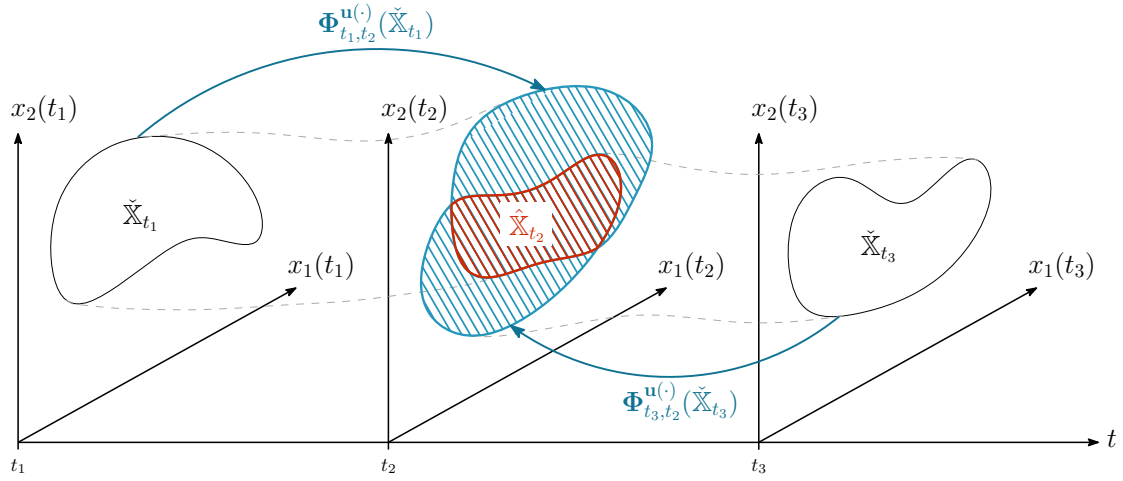


Figure 1: Illustration of one slice of a posterior tube $\hat{\mathbb{X}}(\cdot)$ at time t_2 (red hatched part). This slice is defined as the intersection set $\hat{\mathbb{X}}_{t_2} = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\check{\mathbb{X}}_{t_1}) \cap \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_3,t_2}(\check{\mathbb{X}}_{t_3})$, with $\check{\mathbb{X}}_{t_1}$ and $\check{\mathbb{X}}_{t_3}$ prior states. The flow function $\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t,t_2}$ transports the prior sets from t to t_2, according to the input $\mathbf{u}(\cdot)$ over $[t, t_2]$. The result of each transportation, from t_1 and from t_3, is depicted in blue at time t_2.

Theorem 3. *Given the sampling times of measurements*[1]

$$\mathbb{T} = \{0, \delta, 2\delta, \dots, \bar{k}\delta\} = \{t_0, t_1, t_2, \dots, t_{\bar{k}}\},$$

and a prior tube $\check{\mathbb{X}}(\cdot)$ containing the actual state trajectory $\mathbf{x}(\cdot)$. The posterior tube $\hat{\mathbb{X}}(\cdot)$ can be defined recursively by

$$\begin{aligned}
\overrightarrow{\mathbb{X}}_{t_k} &= \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \cap \bigcap_{\tau\in[t_{k-1},t_k]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\
\overleftarrow{\mathbb{X}}_{t_k} &= \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}) \cap \bigcap_{\tau\in[t_k,t_{k+1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \qquad (10) \\
\hat{\mathbb{X}}_{t_k} &= \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k}
\end{aligned}$$

*with $\overrightarrow{\mathbb{X}}_{t_0} = \check{\mathbb{X}}(t_0)$ and $\overleftarrow{\mathbb{X}}_{t_{\bar{k}}} = \check{\mathbb{X}}(t_{\bar{k}})$. Moreover, when t is not consistent with the sampling times (*i.e. *for $t \in [t_k, t_{k+1}]\backslash\mathbb{T} =]t_k, t_{k+1}[$), then we directly have*

$$\hat{\mathbb{X}}_t = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_k,t}(\hat{\mathbb{X}}_{t_k}).$$

Proof. We have the decomposition:

$$\overrightarrow{\mathbb{X}}_{t_k} \overset{(7)}{=} \bigcap_{\tau\in[0,t_{k-1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \cap \bigcap_{\tau\in[t_{k-1},t_k]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \ . \qquad (11)$$

Now,

$$\begin{aligned}
\bigcap_{\tau\in[0,t_{k-1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) &\overset{(4)}{=} \bigcap_{\tau\in[0,t_{k-1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_{k-1}}(\check{\mathbb{X}}_\tau) \\
&\overset{(5)}{=} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k}\left(\bigcap_{\tau\in[0,t_{k-1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_{k-1}}(\check{\mathbb{X}}_\tau)\right) \\
&\overset{(7)}{=} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}).
\end{aligned} \qquad (12)$$

Combined with Equation (11), we obtain the first part of Equations (10).

Then, for the second part, we have

$$\overleftarrow{\mathbb{X}}_{t_k} \overset{(8)}{=} \bigcap_{\tau\in[t_{k+1},\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \cap \bigcap_{\tau\in[t_k,t_{k+1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \ . \qquad (13)$$

Now,

$$\begin{aligned}
\bigcap_{\tau\in[t_{k+1},\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) &\overset{(4)}{=} \bigcap_{\tau\in[t_{k+1},\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k+1},t_k} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_{k+1}}(\check{\mathbb{X}}_\tau) \\
&\overset{(5)}{=} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k+1},t_k}\left(\bigcap_{\tau\in[t_{k+1},\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_{k+1}}(\check{\mathbb{X}}_\tau)\right) \\
&\overset{(8)}{=} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}),
\end{aligned}$$

which leads to the second part of Equations (10) if combined with Equation (13).

Moreover,

$$\hat{\mathbb{X}}_{t_k} \overset{(9)}{=} \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k},$$

which corresponds to the last part of Equations (10).

Finally

$$\begin{aligned}
\hat{\mathbb{X}}_t &\overset{(6)}{=} \textstyle\bigcap_{\tau\in[0,\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \\
&\overset{(4)}{=} \textstyle\bigcap_{\tau\in[0,\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_k,t} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\
&\overset{(5)}{=} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_k,t}\left(\textstyle\bigcap_{\tau\in[0,\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau)\right) \\
&= \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_k,t}(\hat{\mathbb{X}}_{t_k})
\end{aligned},$$

which concludes the proof. □

2.3. The input $\mathbf{u}(\cdot)$ is uncertain

The previous results can be extended to the case where $\mathbf{u}(\cdot)$ is uncertain but known to be inside a tube $\mathbb{U}(\cdot)$. This allows to consider bounded errors on the system input. In this case, the set flow given by Equation (3) becomes:

$$\mathbf{\Phi}^{\mathbb{U}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) = \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1).$$

[1] Here, δ is the time spacing between measurements and it is assumed to be constant for ease of reading, without loss of generality.

Furthermore, from Equation (6), the smallest tube $\hat{\mathbb{X}}(\cdot)$ for $\mathbf{x}(\cdot)$ consistent with the prior tube $\check{\mathbb{X}}(\cdot)$, the input tube $\mathbb{U}(\cdot)$ and the state equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ is

$$\hat{\mathbb{X}}_t = \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[0,\bar{t}]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau). \tag{14}$$

Remark 4. *Note that for a collection of sets $\mathbb{A}_{ij}$, we have*

$$\bigcup_i \bigcap_j \mathbb{A}_{ij} \subset \bigcap_j \bigcup_i \mathbb{A}_{ij}.$$

Indeed,

$$\begin{array}{rcl} a \in \bigcup_i \bigcap_j \mathbb{A}_{ij} & \Leftrightarrow & \exists i, \forall j, a \in \mathbb{A}_{ij} \\ & \Rightarrow & \forall j, \exists i, a \in \mathbb{A}_{ij} \\ & \Leftrightarrow & a \in \bigcap_j \bigcup_i \mathbb{A}_{ij}. \end{array}$$

As a consequence, we obtain the following inclusion for expressing Equation (14) *(but not an equality):*

$$\begin{array}{rcl} \hat{\mathbb{X}}_t & = & \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[0,\bar{t}]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \\ & \subset & \bigcap_{\tau\in[0,\bar{t}]} \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \\ & = & \bigcap_{\tau\in[0,\bar{t}]} \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau). \end{array}$$

Define the set

$$\overrightarrow{\mathbb{X}}_t = \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[0,t]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \tag{15}$$

that corresponds to the set of all $\mathbf{x}(t)$ consistent with the past (before t) and

$$\overleftarrow{\mathbb{X}}_t = \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t,\bar{t}]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \tag{16}$$

the set of all $\mathbf{x}(t)$ consistent with the future (after t). The exact sequence Theorem extends as follows.

Theorem 5. *Given the sampling times of measurements*

$$\mathbb{T} = \{0, \delta, 2\delta, \ldots, \bar{k}\delta\} = \{t_0, t_1, t_2, \ldots, t_{\bar{k}}\},$$

a prior tube $\check{\mathbb{X}}(\cdot)$ containing the state trajectory $\mathbf{x}(\cdot)$, and a tube $\mathbb{U}(\cdot)$ for the input. The posterior tube $\hat{\mathbb{X}}(\cdot)$ can be defined recursively by

$$\begin{array}{rcl} \overrightarrow{\mathbb{X}}_{t_k} & = & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \cap \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t_{k-1},t_k]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \overleftarrow{\mathbb{X}}_{t_k} & = & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}) \cap \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t_k,t_{k+1}]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \hat{\mathbb{X}}_{t_k} & = & \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k} \end{array} \tag{17}$$

with $\overrightarrow{\mathbb{X}}_{t_0} = \check{\mathbb{X}}(t_0)$ and $\overleftarrow{\mathbb{X}}_{t_{\bar{k}}} = \check{\mathbb{X}}(t_{\bar{k}})$. Moreover, for $t \in [t_k, t_{k+1}]\backslash\mathbb{T}$, we have:

$$\hat{\mathbb{X}}_t = \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_k,t}(\hat{\mathbb{X}}_{t_k}).$$

Proof. From Equation (15), we have

$$\begin{array}{rcl} \overrightarrow{\mathbb{X}}_{t_k} & \overset{(15)}{=} & \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[0,t_k]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ & = & \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \left(\bigcap_{\tau\in[0,t_{k-1}]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \right. \\ & & \left. \cap \bigcap_{\tau\in[t_{k-1},t_k]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \right) \\ & = & \left(\bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[0,t_{k-1}]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \right) \\ & & \cap \left(\bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t_{k-1},t_k]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \right). \end{array} \tag{18}$$

The distribution of the union operator is allowed here since different instants are involved. Indeed, recall that for two sets $\mathbb{A}$ and $\mathbb{B}$, we cannot state that

$$\bigcup_{\mathbf{u}\in\mathbb{U}} \big(\mathbb{A}(\mathbf{u}) \cap \mathbb{B}(\mathbf{u})\big) = \bigcup_{\mathbf{u}\in\mathbb{U}} \mathbb{A}(\mathbf{u}) \cap \bigcup_{\mathbf{u}\in\mathbb{U}} \mathbb{B}(\mathbf{u}).$$

However, this is allowed when we have independence between the components. For instance, if $\mathbb{U}$ is a box of $\mathbb{R}^n$, the above expression is allowed when $\mathbb{A}$ depends on some components of $\mathbb{U}$ and $\mathbb{B}$ of some other components of $\mathbb{U}$.

Now,

$$\begin{array}{rcl} \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[0,t_{k-1}]} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) & \overset{(12)}{=} & \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \\ & = & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}). \end{array}$$

Combined with Equation (18), we obtain the first part of Equations (17). A similar reasoning applies to get the sequence for $\overleftarrow{\mathbb{X}}_{t_k}$ and the expression for $\hat{\mathbb{X}}_t$. □

3. State estimator

The exact sequence suggested by Theorem 3 is valid even if the system is non-linear. Nevertheless, it can be implemented exactly on a computer only in the linear case. This is due to the fact that we do have an expression for the flow. This section shows how an accurate approximation of the exact sequence can be implemented.

3.1. Principle

To use Equation (2) in the case where interval uncertainties exist for time t, we need to introduce the concept of exponential for an interval matrix $[\mathbf{A}]$ which has to be understood with a set-theoretical meaning [26]. It is defined as the smallest interval matrix which encloses all feasible exponentials of $\mathbf{A}$, assuming that $\mathbf{A} \in [\mathbf{A}]$, *i.e.*,

$$e^{[\mathbf{A}]} = \left[\left\{\mathbf{B} \mid \exists \mathbf{A} \in [\mathbf{A}], \mathbf{B} = e^{\mathbf{A}}\right\}\right].$$

Moreover, given an interval matrix $[\mathbf{A}]$ and a set of vectors $\mathbb{X}$, we define the product $[\mathbf{A}] \cdot \mathbb{X}$ as the smallest box

enclosing all feasible products $\mathbf{A}\cdot\mathbf{x}$ assuming that $\mathbf{x}\in\mathbb{X}$ and $\mathbf{A}\in[\mathbf{A}]$, *i.e.*,

$$[\mathbf{A}]\cdot\mathbb{X}=[\{\mathbf{y}\mid\exists\mathbf{A}\in[\mathbf{A}],\ \exists\mathbf{x}\in\mathbb{X},\ \mathbf{y}=\mathbf{A}\cdot\mathbf{x}\}].$$

The following theorem corresponds to an implementation of the exact sequence provided by Theorem 3.

Theorem 6. *Given the sampling times of measurements*

$$\mathbb{T}=\{0,\delta,2\delta,\ldots,\bar{k}\delta\}=\{t_0,t_1,t_2,\ldots,t_{\bar{k}}\},$$

a prior tube $\check{\mathbb{X}}(\cdot)$ *containing the state trajectory* $\mathbf{x}(\cdot)$*, and a piecewise constant tube*

$$\mathbb{U}(\cdot)=\{\mathbf{u}(\cdot)\mid\forall k,\forall t\in[k\delta,(k+1)\delta],\ \mathbf{u}(t)\in[\mathbf{u}]_k\}$$

containing $\mathbf{u}(\cdot)$*, where* $[\mathbf{u}]_k, k\in\{0,\ldots,\bar{k}-1\}$ *is a slice of the tube* $\mathbb{U}(\cdot)$ *as illustrated by Figure 2. We have*

$$\begin{array}{rcl}\overrightarrow{\mathbb{X}}_{t_k} & \subset & \check{\mathbb{X}}_{t_k}\cap\left\{e^{\mathbf{A}\delta}\cdot\overrightarrow{\mathbb{X}}_{t_{k-1}}+\delta e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k-1}\right\}\\ \overleftarrow{\mathbb{X}}_{t_k} & \subset & \check{\mathbb{X}}_{t_k}\cap\left\{e^{-\mathbf{A}\delta}\cdot\overleftarrow{\mathbb{X}}_{t_{k+1}}-\delta e^{-\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_k\right\}\\ \hat{\mathbb{X}}_{t_k} & = & \overrightarrow{\mathbb{X}}_{t_k}\cap\overleftarrow{\mathbb{X}}_{t_k}\end{array}\tag{19}$$

with $\overrightarrow{\mathbb{X}}_{t_0}=\check{\mathbb{X}}_{t_0}$ *and* $\overleftarrow{\mathbb{X}}_{t_{\bar{k}}}=\check{\mathbb{X}}_{t_{\bar{k}}}$.
Moreover for $t\in[t_k,t_{k+1}]\backslash\mathbb{T}$*, we have*

$$\hat{\mathbb{X}}_t=\left\{e^{\mathbf{A}(t-t_k)}\cdot\overrightarrow{\mathbb{X}}_{t_k}+(t-t_k)e^{\mathbf{A}\cdot[0,t-t_k]}\mathbf{B}[\mathbf{u}]_k\right\}.$$

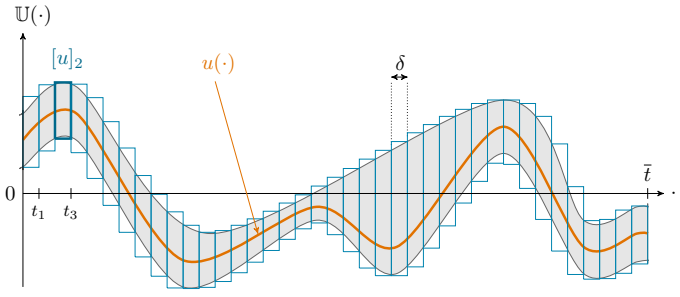


Figure 2: Illustration of a tube $\mathbb{U}(\cdot)$ made of slices. This representation can be used in order to reliably enclose sets of trajectories. For instance, the thick blue box depicts the $[u]_{k=2}$ slice containing the feasible values for $u(t)$, $t\in[t_2,t_3]$.

Remark 7. *The expressions used in the previous formula have to be understood with a set-theoretical meaning. For instance*

$$e^{\mathbf{A}[\delta]}\cdot\mathbb{X}=\left\{\mathbf{y}\mid\exists\mathbf{x}\in\mathbb{X},\exists\delta\in[\delta],\ \mathbf{y}=e^{\mathbf{A}\delta}\cdot\mathbf{x}\right\}.$$

Proof. We first show the inclusion for $\overrightarrow{\mathbb{X}}_{t_k}$. We have

$$\begin{array}{rcll}\overrightarrow{\mathbb{X}}_{t_k} & \overset{(17)}{=} & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}})\cap\bigcup_{\mathbb{U}(\cdot)}\bigcap_{\tau\in[t_{k-1},t_k]}\boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau)\\ & \overset{\text{(Remark 4)}}{\subset} & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}})\cap\bigcap_{\tau\in[t_{k-1},t_k]}\boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau)\\ & \subset & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}})\cap\boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_k,t_k}(\check{\mathbb{X}}_{t_k})\\ & = & \check{\mathbb{X}}_{t_k}\cap\boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}})\\ & = & \check{\mathbb{X}}_{t_k}\cap\left\{e^{\mathbf{A}\delta}\cdot\overrightarrow{\mathbb{X}}_{t_{k-1}}\right.\\ & & \qquad\left.+\int_{t_{k-1}}^{t_k}e^{\mathbf{A}(t_k-\tau)}\mathbf{B}\mathbb{U}(\tau)d\tau\right\}.\end{array}$$

Now, because $\mathbb{U}(\cdot)$ is piecewise constant,

$$\begin{array}{rcl}\int_{t_{k-1}}^{t_k}e^{\mathbf{A}(t_k-\tau)}\mathbf{B}\mathbb{U}(\tau)d\tau & = & \int_{t_{k-1}}^{t_k}e^{\mathbf{A}(t_k-\tau)}\mathbf{B}[\mathbf{u}]_{k-1}d\tau\\ & \subset & \int_{t_{k-1}}^{t_k}e^{\mathbf{A}(t_k-[t_{k-1},t_k])}\mathbf{B}[\mathbf{u}]_{k-1}d\tau\\ & = & \int_{t_{k-1}}^{t_k}e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k-1}d\tau\\ & = & \left(\int_{t_{k-1}}^{t_k}e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}d\tau\right)[\mathbf{u}]_{k-1}\\ & = & \delta e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k-1}.\end{array}\tag{20}$$

Therefore,

$$\overrightarrow{\mathbb{X}}_{t_k}\subset\check{\mathbb{X}}_{t_k}\cap\left\{e^{\mathbf{A}\delta}\cdot\overrightarrow{\mathbb{X}}_{t_{k-1}}+\delta e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k-1}\right\}$$

which corresponds to the first part of Equations (19).

We now show the inclusion for $\overleftarrow{\mathbb{X}}_{t_k}$. We have

$$\begin{array}{rcll}\overleftarrow{\mathbb{X}}_{t_k} & \overset{(17)}{=} & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}})\cap\bigcup_{\mathbb{U}(\cdot)}\bigcap_{\tau\in[t_k,t_{k+1}]}\boldsymbol{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau)\\ & \overset{\text{(Rem. 4)}}{\subset} & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}})\cap\bigcap_{\tau\in[t_k,t_{k+1}]}\boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau)\\ & \subset & \boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}})\cap\boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_k,t_k}(\check{\mathbb{X}}_{t_k})\\ & = & \check{\mathbb{X}}_{t_k}\cap\boldsymbol{\Phi}^{\mathbb{U}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}})\\ & \overset{(2)}{=} & \check{\mathbb{X}}_{t_k}\cap\left\{e^{-\mathbf{A}\delta}\cdot\overleftarrow{\mathbb{X}}_{t_{k+1}}\right.\\ & & \qquad\left.+\int_{t_{k+1}}^{t_k}e^{\mathbf{A}(t_k-\tau)}\mathbf{B}\mathbb{U}(\tau)d\tau\right\}.\end{array}$$

Now, using Equation (20), we obtain

$$\int_{t_{k+1}}^{t_k}e^{\mathbf{A}(t_k-\tau)}\mathbf{B}\mathbb{U}(\tau)d\tau\subset-\delta e^{-\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_k.$$

Therefore,

$$\overleftarrow{\mathbb{X}}_{t_k}\subset\check{\mathbb{X}}_{t_k}\cap\left\{e^{-\mathbf{A}\delta}\cdot\overleftarrow{\mathbb{X}}_{t_{k+1}}-\delta e^{-\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_k\right\}.$$

which corresponds to the second part of Equations (19). The last part is already known from Equations (17). □

3.2. Example: the triple integrator

Before going into the implementation algorithm, we propose the following example for illustrating the method.

Consider the linear system

$$\dot{\mathbf{x}} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_{\mathbf{A}} \mathbf{x} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}_{\mathbf{b}} u$$

equivalent to a triple integrator. We have the flow expressed as

$$\mathbf{\Phi}_{t_1,t_2}^{u(\cdot)}(\mathbf{x}) \overset{(2)}{=} e^{\mathbf{A}(t_2-t_1)}\mathbf{x} + \int_{t_1}^{t_2} e^{\mathbf{A}(t_2-\tau)}\mathbf{b}u(\tau)d\tau,$$

with

$$e^{\mathbf{A}t} = \begin{pmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{pmatrix}.$$

From Theorem 6, we get the following inclusions:

$$\begin{aligned}
\overrightarrow{\mathbb{X}}_{t_k} &\subset \check{\mathbb{X}}_{t_k} \cap \left\{ e^{\mathbf{A}\delta} \cdot \overrightarrow{\mathbb{X}}_{t_{k-1}} + \delta \begin{pmatrix} \frac{[0,\delta]^2}{2} \\ [0,\delta] \\ 1 \end{pmatrix} [u]_{k-1} \right\}, \\
\overleftarrow{\mathbb{X}}_{t_k} &\subset \check{\mathbb{X}}_{t_k} \cap \left\{ e^{-\mathbf{A}\delta} \cdot \overleftarrow{\mathbb{X}}_{t_{k+1}} - \delta \begin{pmatrix} \frac{[0,\delta]^2}{2} \\ [-\delta,0] \\ 1 \end{pmatrix} [u]_{k} \right\}, \\
\hat{\mathbb{X}}_{t_k} &= \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k}.
\end{aligned}$$

Remark 8. *Note that since the system is linear and time invariant (*i.e., $\mathbf{A}$ *does not change with time), we were able to provide a closed form expression for the exponential* $e^{\mathbf{A}t}$*. This time invariance is needed by our method.*

3.3. Implementation

Now that we have defined a reliable enclosure from a bounded input $\mathbb{U}(\cdot)$, that can be numerically represented and guaranteed to enclose $\mathbf{u}(\cdot)$, it remains to reliably compute the sets $\mathbb{X}$. Consider a polygon $\mathcal{P}_k$ which contains $\check{\mathbb{X}}_{t_k}$. We apply the following *polygonal sequence:*

— first, forward in time, for $k \in \{1, \ldots, \bar{k}\}$:

$$\mathcal{P}_k := \mathcal{P}_k \cap \left\{ e^{\mathbf{A}\delta} \cdot \mathcal{P}_{k-1} + \delta e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k-1} \right\},$$

— then, backward in time, for $k \in \{\bar{k}-1, \ldots, 0\}$:

$$\mathcal{P}_k := \mathcal{P}_k \cap \left\{ e^{-\mathbf{A}\delta} \cdot \mathcal{P}_{k+1} - \delta e^{-\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k} \right\},$$

— finally, between the sampling times, $k \in \{0, \ldots, \bar{k}-1\}$:

$$\mathcal{P}_{k:k+1} = \left\{ e^{\mathbf{A}[0,\delta]} \cdot \mathcal{P}_k + [0,\delta] e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k} \right\}.$$

From Theorem 6, we know that for all $t_k \in \mathbb{T}$,

$$\begin{cases} \hat{\mathbb{X}}_{t_k} \subset \mathcal{P}_k, \\ \forall t \in [t_k, t_{k+1}],\ \hat{\mathbb{X}}_t \subset \mathcal{P}_{k:k+1}. \end{cases}$$

An illustration of the polygons $\mathcal{P}_k$, $\mathcal{P}_{k:k+1}$, $\mathcal{P}_{k+1}$ implementing a slice of a tube $\hat{\mathbb{X}}(\cdot)$ is given by Figure 3.

The implementation is close to the exact sequence given by Theorem 3. As a consequence, it is not needed in practice to apply several times the polygonal sequence to obtain an accurate enclosure.

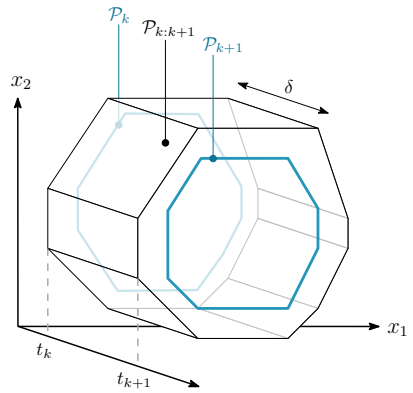


Figure 3: A polygon-slice enclosing the tube $\hat{\mathbb{X}}(\cdot)$ over $[t_k, t_{k+1}]$. The *gates* of the slice are the polygons $\mathcal{P}_k$ and $\mathcal{P}_{k+1}$, blue painted.

Remark 9. *When the state estimator is numerically implemented, rounding errors occur. Due to the fundamental theorem of interval analysis [27], an interval implementation with outward rounding makes it possible to get a guaranteed approximation of the exact set. However, it will introduce a small pessimism. The accuracy of the approximation can be made as small as desired if we use a multi-precision interval arithmetic library such as MPFI [28].*

3.4. Measurements at uncertain times

In this paper, an observation made at some time t consistent with the sampling times $\mathbb{T}$ is processed as a restriction on the $\check{\mathbb{X}}(\cdot)$ prior sets. However, in practice, the time of measurement t is not consistent with $\mathbb{T}$. This makes the problem much more difficult if the guarantee is required [29, 30]. Interval analysis can help to deal with this type of estimation [31].

Define as t_k the largest $t_k \in \mathbb{T}$ such that $t_k \leqslant t$. If $[\mathbf{y}]$ is a bounded measurement, with $\mathbf{x}_t \in [\mathbf{y}]$, then we have:

$$\mathbf{x}_{t_k} \in e^{-\mathbf{A}(t-t_k)} \cdot [\mathbf{y}] - (t-t_k) e^{-\mathbf{A}\cdot[0,t-t_k]}\mathbf{B}[\mathbf{u}]_k.$$

If now, t is uncertain, but known to belong to $[t_k, t_{k+1}]$, then we have $t - t_k \in [0, \delta]$. Therefore

$$\mathbf{x}_{t_k} \in e^{-\mathbf{A}[0,\delta]} \cdot [\mathbf{y}] - [0,\delta] e^{-\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_k$$

where $\delta = t_{k+1} - t_k$.

Finally, if the uncertainty covers several slices, *i.e.* $t \in [t_{k_a}, t_{k_b}]$, we get

$$\mathbf{x}_{t_{k_1}} \in e^{-\mathbf{A}[0,\delta]} \cdot [\mathbf{y}] - [0,\delta] e^{-\mathbf{A}\cdot[0,\delta]}\mathbf{B} \cdot [\mathbf{u}]_{k_a:k_b},$$

where $\delta = t_{k_b} - t_{k_a}$ and $[\mathbf{u}]_{k_a:k_b} = \bigsqcup_{k\in[k_a,k_b]} [\mathbf{u}]_k$.

This is similar to the implementation used in [32] for reliably handling time uncertainties.

4. Test case

In this section, we provide an application of the proposed implementation for state estimation. Let us consider

a car moving on a line over $[0, \overline{t}] = [0, 10]$. The system is described by

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - x_2 + u \end{cases},$$

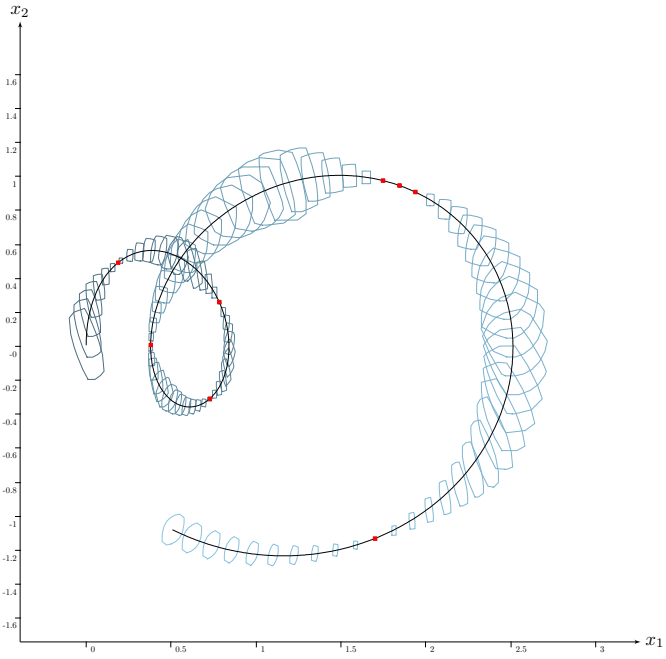
where x_1 corresponds to the position of the car, x_2 to the speed. The initial condition $\mathbf{x}(0) = \mathbf{0}$ is considered unknown. We assume that the thrust u satisfies $u \in \cos(t) + \sin(t/3) + t/10 + [-0.1, 0.1]$. The state estimation is performed with a time step δ, which defines the slicing of the input tube $\mathbb{U}(\cdot)$ and the state tube $\hat{\mathbb{X}}(\cdot)$. Measurements of the state vector $(\tilde{y}_1, \tilde{y}_2)$ are provided at times $t_i \in [0, \overline{t}]$ as given in Table 1. Note that the t_i's are not necessarily consistent with the sampling times $k\delta$ we use for the computation. The measurement errors are assumed to be 0.01, *i.e.*, $\forall t_i, \mathbf{x}(t_i) \in \tilde{\mathbf{y}}(t_i) + [-0.01, 0.01]^2$.

Table 1: State observation vectors $\tilde{\mathbf{y}}(t_i)$.

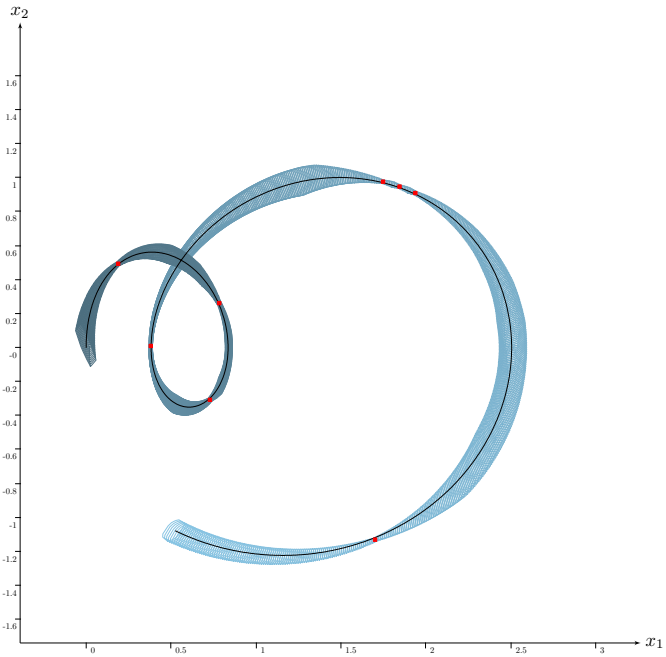
t_i	2/3	1.9	2.99	4.33	6.4	6.5	6.6	9.0
$\tilde{y}_1$	0.188	0.783	0.728	0.380	1.747	1.844	1.937	1.700
$\tilde{y}_2$	0.493	0.261	−0.308	0.009	0.976	0.947	0.909	−1.121

The polygonal enclosure of the state trajectory is computed forward and backward in time, allowing to enclose the unknown initial condition $\mathbf{x}(0)$. Results are illustrated in Figure 4 for $\delta = 0.1$ and $\delta = 0.01$.

Remark 10. *The envelop that is computed for different values of δ converges towards the exact envelop as shown by Theorem 3. We observe in Figures 5–6 that no significant improvement is observed if we decrease the value of δ, which illustrates a fast convergence with respect to the sampling time of the tube. The computation time is 0.2 second for $\delta = 0.1$ and 2 seconds for $\delta = 0.01$. Note also that when we increase the dimension of the state space, our method requires too much computation if we still keep a small sampling time δ. However, if we accept some pessimism in the approximation, with values for δ not too small, or if the number of measurements is large enough, we may still be able to use our approach to get an accurate and guaranteed approximation of the solution set.*



(a) $\delta = 0.1$. Computation time: 0.2s.



(b) $\delta = 0.01$. Computation time: 2s.

Figure 4: State trajectory, depicted in black, and approximated by a polygonal sequence. The eight bounded observations $[\mathbf{y}](t_i)$ are depicted in red. The figures illustrate two sampling times. The $\delta = 0.01$ approximation involves more polygons, and thus a more accurate enclosure of the state trajectory.

5. Conclusion

In this paper, we have proposed a new state estimator to approximate the state of a continuous-time linear system in a set-membership context, with a set of discrete measurements. The method is both

— accurate, since it is a direct extension of the exact polygonal method used for discrete time systems;

— guaranteed, because the propagation of uncertainties over time is made rigorously by using interval analysis.

If we assume that the sampling time δ is infinitely small and that the computer computes exactly with real numbers instead of floating point numbers, then our method can be qualified as *exact* since it does not introduce any pessimism and does not lose any feasible values.

The proposed combination of a polygonal approach with

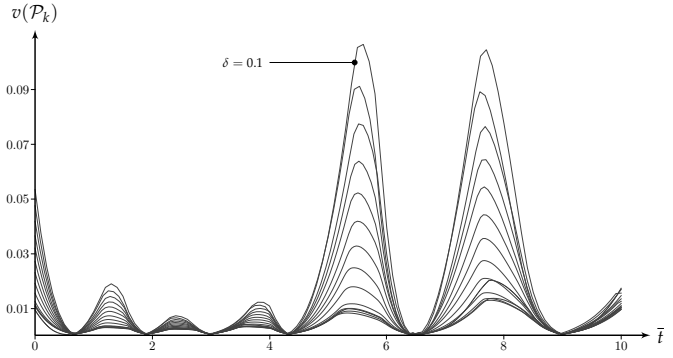


Figure 5: Area of each 2d polygon $\mathcal{P}_k$ computed with several time steps $\delta = 0.1 \ldots 0.0001$. The smallest polygons correspond to the measurement times given in Table 1. We observe that the polygonal enclosure rapidly converges towards the exact envelop, corresponding to an infinitely small time step δ. This is also illustrated by Figure 6.

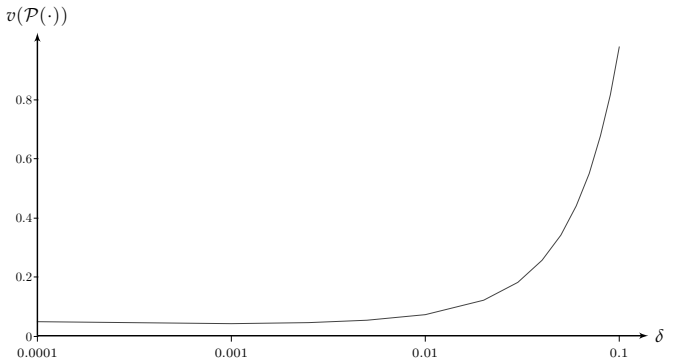


Figure 6: Volume of polygonal tubes obtained for several time steps $\delta = 0.1 \ldots 0.0001$ (logarithmic scale).

interval computations is new in this context. We believe that it is the only way to obtain an estimation that is both accurate and guaranteed. In addition, the same approach allows us to handle rigorously time uncertainties outside the sampling times on which the estimation set is implemented.

It must be underlined that our approach cannot be extended directly to non-linear dynamical systems. However, it is often possible to extract linear differential equations from the dynamics. The method developed here can then be used to deal with these equations, within a constraint propagation process. This will be our main objective for forthcoming work.

Supplementary materials including the sources of the application are available on http://simon-rohou.fr/research/linobs/. This work has been implemented in the Codac library [33], that provides tools for computing sets of trajectories.

Acknowledgments

This work was supported by the French Agence Nationale de la Recherche (ANR) [grant number ANR-16-CE33-0024].

References

[1] R. E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the AMSE, Part D, Journal of Basic Engineering 82 (1960) 35–45.

[2] L. Ljung, Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems, IEEE Transactions on Automatic Systems 24 (1979) 36–50.

[3] F. C. Schweppe, Recursive state estimation: unknown but bounded errors and system inputs, "IEEE Transactions on Automatic Control" 13 (1) (1968) 22–28.

[4] E. Fogel, Y. F. Huang, On the value of information in system identification - bounded noise case, Automatica 18 (2) (1982) 229–238.

[5] M. Milanese, G. Belforte, Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded errors: Linear families of models and estimators, "IEEE Transactions on Automatic Control" 27 (2) (1982) 408–414.

[6] G. Belforte, B. Bona, V. Cerone, Parameter estimation algorithms for a set-membership description of uncertainty, Automatica 26 (5) (1990) 887–898.

[7] A. Kurzhanski, I. Valyi, Ellipsoidal Calculus for Estimation and Control, Birkhäuser, Boston, MA, 1997.

[8] C. Durieu, B. Polyak, E. Walter, Trace versus determinant in ellipsoidal outer bounding with application to state estimation, in: Proceedings of the 13th IFAC World Congress, Vol. I, San Francisco, CA, 1996, pp. 43–48.

[9] J. Xiong, C. Jauberthie, L. Travé-Massuyes, F. L. Gall, Fault detection using interval kalman filtering enhanced by constraint propagation, in: Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, IEEE, 2013, pp. 490–495.

[10] M. Mustafa, A. Stancu, N. Delanoue, E. Codres, Guaranteed SLAM; An Interval Approach, Robotics and Autonomous Systems 100 (2018) 160–170.

[11] T.Alamo, J. B. ad E.F.Camachoa, Guaranteed state estimation by zonotopes, Automatica 41 (6) (2005) 1035–1043.

[12] C. Combastel, A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes, in: CDC-ECC '05, 2005.

[13] V. T. Hieu, C. Stoica, T. Alamo, E. Camacho, D. Dumur, Zonotopic guaranteed state estimation for uncertain systems, Automatica 49 (11) (2013) 3418–3424.

[14] L. Chisci, A. Garulli, G. Zappa, Recursive state bounding by parallelotopes, Automatica 32 (7) (1996) 1049–1055.

[15] J. Wan, S. Sharma, R. Sutton, Guaranteed state estimation for nonlinear discrete-time systems via indirectly implemented polytopic set computation, IEEE Transactions on Automatic Control 63 (12) (2018) 4317–4322.

[16] E. Walter, H. Piet-Lahanier, Exact recursive polyhedral description of the feasible parameter set for bounded-error models, "IEEE Transactions on Automatic Control" 34 (8) (1989) 911–915.

[17] F. Mazenc, M. Kieffer, E. Walter, Interval observers for continuous-time linear systems, ACC 2012, Jun 2012, Montreal, Canada.

[18] T. Raissi, D. Efimov, A. Zolghadri, Interval state estimation for a class of nonlinear systems, IEEE Transactions on Automatic Control 57 (1) (2012) 260–265.

[19] A. Rauh, J. Kersten, H. Aschemann, Techniques for verified reachability analysis of quasi-linear continuous-time systems, in: 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR), 2019, pp. 18–23. doi:10.1109/MMAR.2019.8864648.

[20] D. Efimov, E. Fridman, A. Polyakov, W. Perruquetti, J.-P. Richard, On design of interval observers with sampled measurement, Systems and Control Letters 96 (2016) 158 – 164. doi:https://doi.org/10.1016/j.sysconle.2016.08.002.
URL http://www.sciencedirect.com/science/article/pii/S0167691116301086

[21] A. B. Kurzhanski, T. F. Filippova, On the Theory of Trajectory Tubes - A Mathematical Formalism for Uncertain Dynamics, Viability and Control, in: A. B. Kurzhanski (Ed.), Advances in Nonlinear Dynamics and Control: A Report from Russia, Birkhäuser Boston, Boston, MA, 1993, pp. 122–188. doi:10.1007/978-1-4612-0349-0_4.
URL http://link.springer.com/10.1007/978-1-4612-0349-0_4

[22] B. A. Davey, H. A. Priestley, Introduction to Lattices and Order, Cambridge University Press, (ISBN 0521784514), 2002.

[23] D. P. Bertsekas, I. B. Rhodes, Recursive state estimation for a set-membership description of uncertainty, "IEEE Transactions on Automatic Control" 16 (2) (1971) 117–128.

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

[25] J. Aubin, H. Frankowska, Set-Valued Analysis, Birkhäuser, Boston, 1990.

[26] A. Goldsztejn, A. Neumaier, On the exponentiation of interval matrices, Reliable Computing 20 (2014) 52–72.

[27] R. Moore, Methods and Applications of Interval Analysis, Studies in Applied and Numerical Mathematics, Society for Industrial and Applied Mathematics, 1979.
URL https://books.google.fr/books?id=WYjD2-R2zMgC

[28] N. Revol, F. Rouillier, Motivations for an arbitrary precision interval arithmetic and the MPFI library, Reliable Computing 11 (4) (2002) 275–290.

[29] V. Cerone, Feasible parameter set for linear models with bounded errors in all variables, Automatica 29 (6) (1993) 1551–1555.

[30] S. M. Veres, J. P. Norton, Parameter-bounding algorithms for linear errors-in-variables models, in: M. Milanese, J. Norton, H. Piet-Lahanier, E. Walter (Eds.), Bounding Approaches to System Identification, Plenum, New York, NY, 1996, pp. 275–288.

[31] L. Jaulin, E. Walter, Guaranteed bounded-error parameter estimation for nonlinear models with uncertain experimental factors, Automatica 35 (5) (1999) 849–856.

[32] S. Rohou, L. Jaulin, L. Mihaylova, F. Le Bars, S. M. Veres, Reliable non-linear state estimation involving time uncertainties, Automatica 93 (2018) 379–388. doi:https://doi.org/10.1016/j.automatica.2018.03.074.

[33] S. Rohou, B. Desrochers, et al., The Codac library – Constraint-programming for robotics, http://codac.io (2021).