Exact bounded-error continuous-time linear state estimator

Simon Rohou, Luc Jaulin

ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France

AID Nov 2022
22nd November 2022

Section 1

Introduction

Introduction

Consider the system:

- linear, continuous, time-invariant
- measurements are collected at discrete times
- all errors are assumed to be intervals
- states are known to belong to some prior sets, possibly infinite if nothing is known *a priori*.

Introduction

Consider the system:

- linear, continuous, time-invariant
- measurements are collected at discrete times
- all errors are assumed to be intervals
- states are known to belong to some prior sets, possibly infinite if nothing is known *a priori*.

Goal – obtain a state estimator:

- consistent with:
  - the state equation
  - the observations at discrete times (possibly uncertain)
  - the related errors
- guaranteed to *continuously* enclose the state vectors
- and **exact**, since it does not introduce pessimism and does not lose any consistent state

Introduction

State-of-the-art

Linear systems usually treated in the **discrete case**:

- **A new approach to linear filtering and prediction problems**
R. E. Kalman, *Tr. of the AMSE, Journal of Basic Engineering*, 1960

- **Recursive state estimation: unknown but bounded errors..**
F. C. Schweppe, *IEEE TAC*, 1968

Introduction

State-of-the-art

Linear systems usually treated in the **discrete case**:

- **A new approach to linear filtering and prediction problems**
  R. E. Kalman, *Tr. of the AMSE, Journal of Basic Engineering*, 1960

- **Recursive state estimation: unknown but bounded errors..**
  F. C. Schweppe, *IEEE TAC*, 1968

In the **continuous case**..

- **Techniques for verified reachability analysis of quasi-linear continuous-time systems**
  A. Rauh, J. Kersten, H. Aschemann, *Intern. Conf. on MMAR*, 2019

→ conservatism/overestimations mainly due to the fact that necessary conditions are used, such as positivity, or wrapping effects.

Section 2

Exact sequence

Exact sequence

Flow expression

Consider the linear time-invariant dynamical system

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

The system is linear
$\rightarrow$ an analytical expression for the flow is given by

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

Exact sequence

Continuous recursive state estimation

For $t \in [0, \bar{t}]$, the state trajectory $\mathbf{x}(\cdot)$ is known to be inside the prior tube $\check{\mathbb{X}}(\cdot)$.

Goal: 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.

$\rightarrow$ Extension to continuous time systems of the state estimator proposed in:

Recursive state estimation for a set-membership description..
D. P. Bertsekas, I. B. Rhodes. *IEEE TAC*, 1971

Exact sequence

Continuous recursive state estimation

Recursive expression of the states:

$$\mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1) \tag{3}$$

Exact sequence

Continuous recursive state estimation

Recursive expression of the states:

$$\mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1) \tag{3}$$

The flow can be extended to sets:

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) = \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_1 \in \mathbb{X}_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1)\right\} \tag{4}$$

Exact sequence

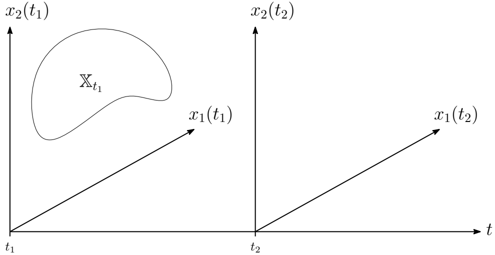
Continuous recursive state estimation

Recursive expression of the states:

$$\mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1) \tag{3}$$

The flow can be extended to sets:

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) = \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_1 \in \mathbb{X}_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1)\right\} \tag{4}$$



Exact sequence

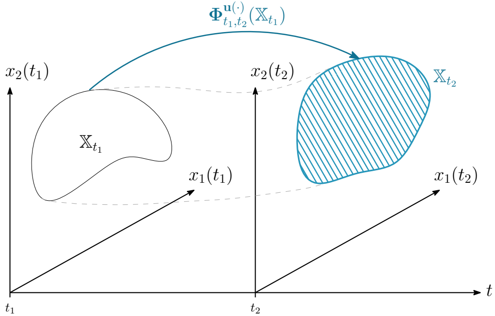
Continuous recursive state estimation

Recursive expression of the states:

$$\mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1) \tag{3}$$

The flow can be extended to sets:

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) = \left\{\mathbf{x}_2 \mid \exists \mathbf{x}_1 \in \mathbb{X}_1,\ \mathbf{x}_2 = \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbf{x}_1)\right\} \tag{4}$$



Exact sequence

Flow set properties

Chasles property:

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_3}(\mathbb{X}_1) \quad = \quad \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_2,t_3} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) \tag{5}$$

Exact sequence

Flow set properties

Chasles property:

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_3}(\mathbb{X}_1) \quad = \quad \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_2,t_3} \circ \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}_1) \tag{5}$$

Automorphism property: given two sets $\mathbb{X}^a_1, \mathbb{X}^b_1$,

$$\mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^a_1 \cap \mathbb{X}^b_1) \quad = \quad \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^a_1) \cap \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_1,t_2}(\mathbb{X}^b_1) \tag{6}$$

Exact sequence

Posterior state tube $\hat{\mathbb{X}}(\cdot)$

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

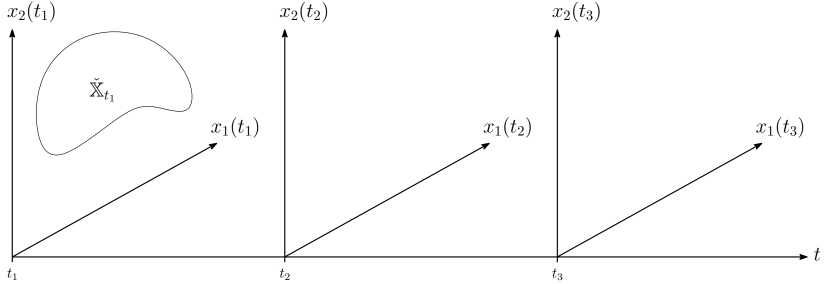
$$\hat{\mathbb{X}}_t = \bigcap_{\tau \in [0,\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \tag{7}$$

Exact sequence

Posterior state tube $\hat{\mathbb{X}}(\cdot)$

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

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

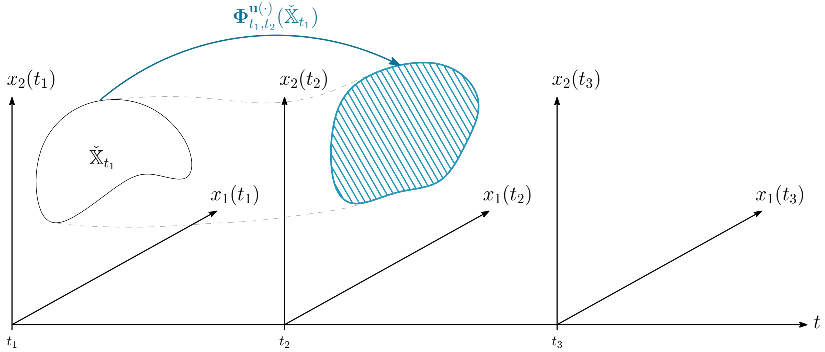


Exact sequence

Posterior state tube $\hat{\mathbb{X}}(\cdot)$

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

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

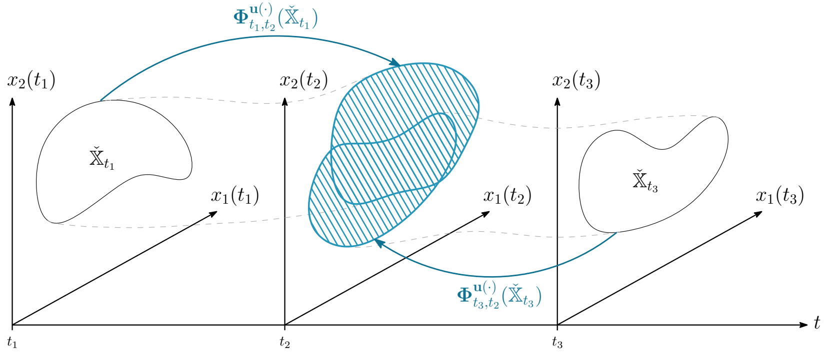


Exact sequence

Posterior state tube $\hat{\mathbb{X}}(\cdot)$

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

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

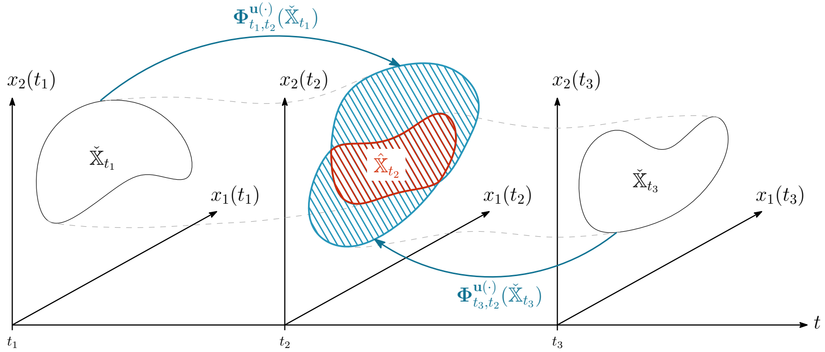


Exact sequence

Posterior state tube $\hat{\mathbb{X}}(\cdot)$

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

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



Exact sequence

State sets consistent with the future/past

Define the set $\overrightarrow{\mathbb{X}}_t$ as the set of all $\mathbf{x}(t)$ consistent with the past (before t)

$$\overrightarrow{\mathbb{X}}_t = \bigcap_{\tau \in [0,t]} \mathbf{\Phi}_{\tau,t}^{\mathbf{u}(\cdot)}(\check{\mathbb{X}}_\tau) \tag{8}$$

Exact sequence

State sets consistent with the future/past

Define the set $\overrightarrow{\mathbb{X}}_t$ as the set of all $\mathbf{x}(t)$ consistent with the past (before t)

$$\overrightarrow{\mathbb{X}}_t = \bigcap_{\tau \in [0,t]} \mathbf{\Phi}_{\tau,t}^{\mathbf{u}(\cdot)}(\check{\mathbb{X}}_\tau) \tag{8}$$

Define the set $\overleftarrow{\mathbb{X}}_t$ as the set of all $\mathbf{x}(t)$ consistent with the future (after t)

$$\overleftarrow{\mathbb{X}}_t = \bigcap_{\tau \in [t,\bar{t}]} \mathbf{\Phi}_{\tau,t}^{\mathbf{u}(\cdot)}(\check{\mathbb{X}}_\tau) \tag{9}$$

Exact sequence

State sets consistent with the future/past

Define the set $\overrightarrow{\mathbb{X}}_t$ as the set of all $\mathbf{x}(t)$ consistent with the past (before t)

$$\overrightarrow{\mathbb{X}}_t = \bigcap_{\tau\in[0,t]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \tag{8}$$

Define the set $\overleftarrow{\mathbb{X}}_t$ as the set of all $\mathbf{x}(t)$ consistent with the future (after t)

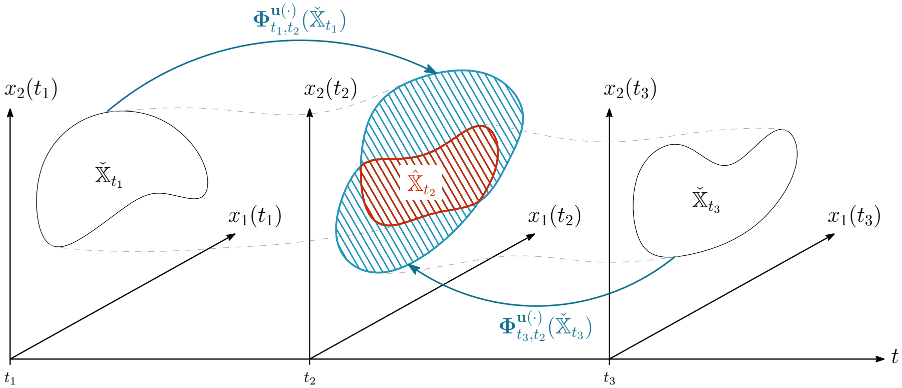
$$\overleftarrow{\mathbb{X}}_t = \bigcap_{\tau\in[t,\bar{t}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t}(\check{\mathbb{X}}_\tau) \tag{9}$$

Intersection at t:

$$\hat{\mathbb{X}}_t = \overrightarrow{\mathbb{X}}_t \cap \overleftarrow{\mathbb{X}}_t \tag{10}$$

Exact sequence

State sets consistent with the future/past



Exact sequence

Recursive algorithm

Given the sampling times of measurements

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

and a prior tube $\check{\mathbb{X}}(\cdot)$ containing the actual state trajectory $\mathbf{x}(\cdot)$.

Exact sequence

Recursive algorithm

Given the sampling times of measurements

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

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

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

Exact sequence

Recursive algorithm

Given the sampling times of measurements

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

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

$$\begin{array}{rcl} \overrightarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \cap \bigcap\limits_{\tau\in[t_{k-1},t_k]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \overleftarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}) \cap \bigcap\limits_{\tau\in[t_k,t_{k+1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \end{array} \quad (11)$$

Exact sequence

Recursive algorithm

Given the sampling times of measurements

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

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

$$\begin{array}{rcl}
\overrightarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \cap \bigcap\limits_{\tau\in[t_{k-1},t_k]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\
\overleftarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}) \cap \bigcap\limits_{\tau\in[t_k,t_{k+1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\
\hat{\mathbb{X}}_{t_k} & = & \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k}
\end{array} \tag{11}$$

Exact sequence

Recursive algorithm

Given the sampling times of measurements

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

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

$$\begin{array}{rcl} \overrightarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \cap \bigcap_{\tau\in[t_{k-1},t_k]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \overleftarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}) \cap \bigcap_{\tau\in[t_k,t_{k+1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \hat{\mathbb{X}}_{t_k} & = & \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k} \end{array} \tag{11}$$

with $\overrightarrow{\mathbb{X}}_{t_0} = \check{\mathbb{X}}(t_0)$ and $\overleftarrow{\mathbb{X}}_{t_{\bar{k}}} = \check{\mathbb{X}}(t_{\bar{k}})$.

Exact sequence

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

Extension to the case where $\mathbf{u}(\cdot)$ is uncertain but known to be inside a tube $\mathbb{U}(\cdot)$. The set flow becomes:

$$\mathbf{\Phi}_{t_1,t_2}^{\mathbb{U}(\cdot)}(\mathbb{X}_1) = \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \mathbf{\Phi}_{t_1,t_2}^{\mathbf{u}(\cdot)}(\mathbb{X}_1). \tag{12}$$

This allows to consider bounded errors on the system input.

Exact sequence

Recursive algorithm ($\mathbf{u}(\cdot)$ uncertain, $\mathbf{u}(\cdot) \in \mathbb{U}(\cdot)$)

Given the sampling times of measurements

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

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

$$\begin{array}{rcl} \overrightarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \cap \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t_{k-1},t_k]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \overleftarrow{\mathbb{X}}_{t_k} & = & \mathbf{\Phi}^{\mathbb{U}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}) \cap \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t_k,t_{k+1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \hat{\mathbb{X}}_{t_k} & = & \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k} \end{array} \quad (13)$$

with $\overrightarrow{\mathbb{X}}_{t_0} = \check{\mathbb{X}}(t_0)$ and $\overleftarrow{\mathbb{X}}_{t_{\bar{k}}} = \check{\mathbb{X}}(t_{\bar{k}})$.

Exact sequence

Recursive algorithm ($\mathbf{u}(\cdot)$ uncertain, $\mathbf{u}(\cdot) \in \mathbb{U}(\cdot)$)

Given the sampling times of measurements

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

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

$$\begin{aligned} \overrightarrow{\mathbb{X}}_{t_k} &= \mathbf{\Phi}^{\mathbb{U}(\cdot)}_{t_{k-1},t_k}(\overrightarrow{\mathbb{X}}_{t_{k-1}}) \cap \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t_{k-1},t_k]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \overleftarrow{\mathbb{X}}_{t_k} &= \mathbf{\Phi}^{\mathbb{U}(\cdot)}_{t_{k+1},t_k}(\overleftarrow{\mathbb{X}}_{t_{k+1}}) \cap \bigcup_{\mathbf{u}(\cdot)\in\mathbb{U}(\cdot)} \bigcap_{\tau\in[t_k,t_{k+1}]} \mathbf{\Phi}^{\mathbf{u}(\cdot)}_{\tau,t_k}(\check{\mathbb{X}}_\tau) \\ \hat{\mathbb{X}}_{t_k} &= \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k} \end{aligned} \tag{13}$$

with $\overrightarrow{\mathbb{X}}_{t_0} = \check{\mathbb{X}}(t_0)$ and $\overleftarrow{\mathbb{X}}_{t_{\bar{k}}} = \check{\mathbb{X}}(t_{\bar{k}})$.

Exact sequence, valid even for non-linear systems

Section 3

State estimator

State estimator

Exact formulation

Exact sequence, valid even for non-linear systems

- depends on the flow $\mathbf{\Phi}$
- therefore, can be implemented exactly on a computer only in the linear case

State estimator

Exact formulation

Exact sequence, valid even for non-linear systems

- depends on the flow $\mathbf{\Phi}$
- therefore, can be implemented exactly on a computer only in the linear case

For linear systems, the flow is analytically given by:

$$\mathbf{\Phi}_{t_1,t_2}^{\mathbf{u}(\cdot)}(\mathbf{x}) = e^{\mathbf{A}(t_2-t_1)}\mathbf{x} + \int_{t_1}^{t_2} e^{\mathbf{A}(t_2-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau. \qquad (14)$$

State estimator

Exponential of an interval matrix

Need to compute the exponential of an interval matrix $[\mathbf{A}]$ which has to be understood with a set-theoretical meaning:

$$e^{[\mathbf{A}]} = \left[\left\{\mathbf{B} \mid \exists\mathbf{A} \in [\mathbf{A}], \mathbf{B} = e^{\mathbf{A}}\right\}\right] \tag{15}$$

State estimator

Exponential of an interval matrix

Need to compute the exponential of an interval matrix $[\mathbf{A}]$ which has to be understood with a set-theoretical meaning:

$$e^{[\mathbf{A}]} = \left[\left\{\mathbf{B} \mid \exists \mathbf{A} \in [\mathbf{A}], \mathbf{B} = e^{\mathbf{A}}\right\}\right] \quad (15)$$

We also define the product:

$$[\mathbf{A}] \cdot \mathbb{X} = [\{\mathbf{y} \mid \exists \mathbf{A} \in [\mathbf{A}],\ \exists \mathbf{x} \in \mathbb{X},\ \mathbf{y} = \mathbf{A} \cdot \mathbf{x}\}] \quad (16)$$

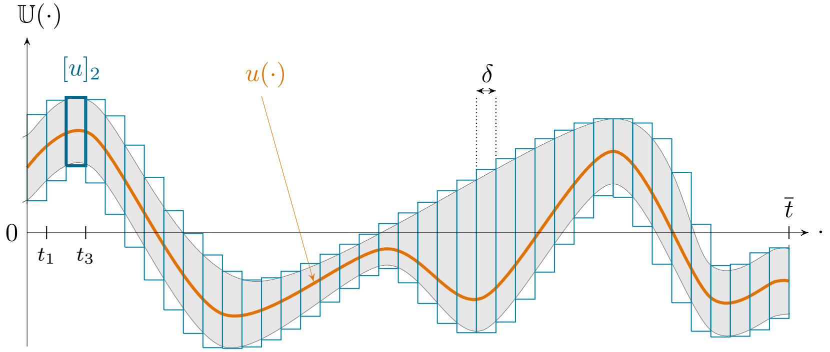
State estimator

Computer implementation of $\mathbb{U}(\cdot)$

We now consider a piecewise constant tube containing $\mathbf{u}(\cdot)$:

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

where $[\mathbf{u}]_k, k \in \{0, \ldots, \bar{k}-1\}$ is a slice of the tube $\mathbb{U}(\cdot)$.



State estimator

Exact formulation for the linear case

Given the sampling times of measurements

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

a prior tube $\check{\mathbb{X}}(\cdot)$ containing the state trajectory $\mathbf{x}(\cdot)$, and a piecewise constant tube $\mathbb{U}(\cdot)$ containing $\mathbf{u}(\cdot)$. We have

$$\begin{array}{rcl} \overrightarrow{\mathbb{X}}_{t_k} & \subset & \check{\mathbb{X}}_{t_k} \cap \left\{ e^{\mathbf{A}\delta} \cdot \overrightarrow{\mathbb{X}}_{t_{k-1}} + \delta e^{\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k-1} \right\} \\ \overleftarrow{\mathbb{X}}_{t_k} & \subset & \check{\mathbb{X}}_{t_k} \cap \left\{ e^{-\mathbf{A}\delta} \cdot \overleftarrow{\mathbb{X}}_{t_{k+1}} - \delta e^{-\mathbf{A}\cdot[0,\delta]}\mathbf{B}[\mathbf{u}]_{k} \right\} \\ \hat{\mathbb{X}}_{t_k} & = & \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k} \end{array} \qquad (17)$$

with $\overrightarrow{\mathbb{X}}_{t_0} = \check{\mathbb{X}}_{t_0}$ and $\overleftarrow{\mathbb{X}}_{t_{\bar{k}}} = \check{\mathbb{X}}_{t_{\bar{k}}}$.

State estimator

Exact implementation: polygonal sequence

We have defined a reliable enclosure from a bounded input $\mathbb{U}(\cdot)$, that can be numerically represented and guaranteed to enclose $\mathbf{u}(\cdot)$.

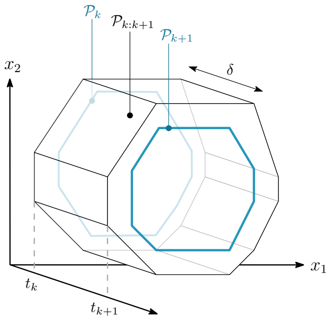
It remains to reliably compute the sets $\mathbb{X}$.

State estimator

Exact implementation: polygonal sequence

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 $\mathbb{X}$.



$\rightarrow$ We use polygons

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

State estimator

Exact implementation: polygonal sequence

We apply the following *polygonal sequence:*

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

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

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

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

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

$$\hat{\mathbb{X}}_{t_k} = \overrightarrow{\mathbb{X}}_{t_k} \cap \overleftarrow{\mathbb{X}}_{t_k}.$$

State estimator

Exact implementation: polygonal sequence

We apply the following *polygonal sequence:*

- first, forward in time, for $k \in \{1, \dots, \bar{k}\}$:

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

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

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

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

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

Section 4

Test case

Test case

Application: moving 1d car

Consider the linear system:

$$\left\{\begin{array}{rcl}\dot{x}_1 & = & x_2 \\ \dot{x}_2 & = & -x_1 - x_2 + u\end{array}\right., \tag{18}$$

and the input bounded as:

$$u \in \cos(t) + \sin(t/3) + t/10 + [-0.1, 0.1]. \tag{19}$$

The initial condition is considered unknown. Its actual value is:

$$\mathbf{x}(0) = \mathbf{0}. \tag{20}$$

Test case

State observations

Direct measurements of the state vector $(\tilde{y}_1, \tilde{y}_2)$ are provided at times $t_i \in [0, \bar{t}]$ as given below:

Table: 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

- t_i's are not necessarily consistent with the sampling times $k\delta$
- measurement errors are assumed to be 0.01, *i.e.*, $\forall t_i, \mathbf{x}(t_i) \in \tilde{\mathbf{y}}(t_i) + [-0.01, 0.01]^2$.

Test case

State observations

Direct measurements of the state vector $(\tilde{y}_1, \tilde{y}_2)$ are provided at times $t_i \in [0, \bar{t}]$ as given below:

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

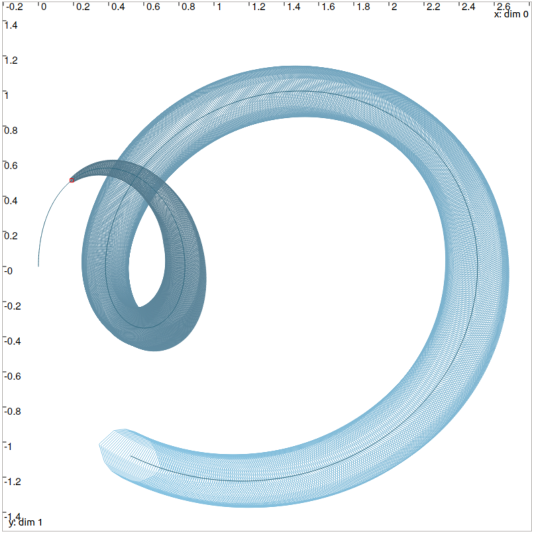
t_i	$^2/_3$	1.9	2.99	4.33	6.4	6.5	6.6	9.0
$\tilde{y}_1$	0.188	0.783	0.728	0.380	1.747	1.844	1.937	1.700
$\tilde{y}_2$	0.493	0.261	-0.308	0.009	0.976	0.947	0.909	-1.121

- t_i's are not necessarily consistent with the sampling times $k\delta$
- measurement errors are assumed to be 0.01, *i.e.*, $\forall t_i, \mathbf{x}(t_i) \in \tilde{\mathbf{y}}(t_i) + [-0.01, 0.01]^2$.

$\rightarrow$ Restrictions on the prior tube $\mathcal{P}(\cdot)$, *i.e.* $\check{\mathbb{X}}(\cdot)$

Test case

Resulting polygonal envelop



Forward computation

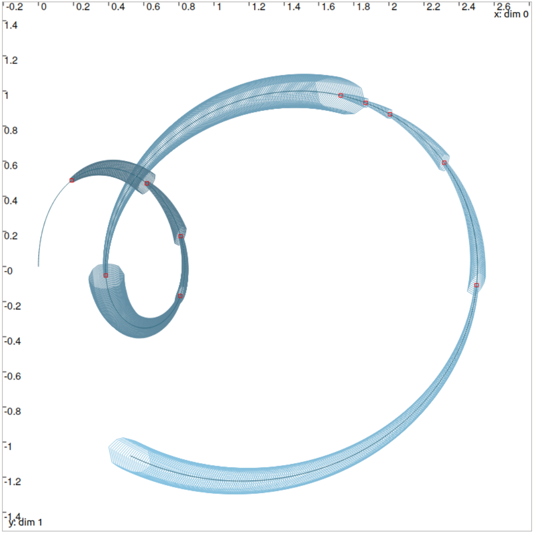
One observation

$\delta = 0.01$

Computation time: $\sim$2s

Test case

Resulting polygonal envelop

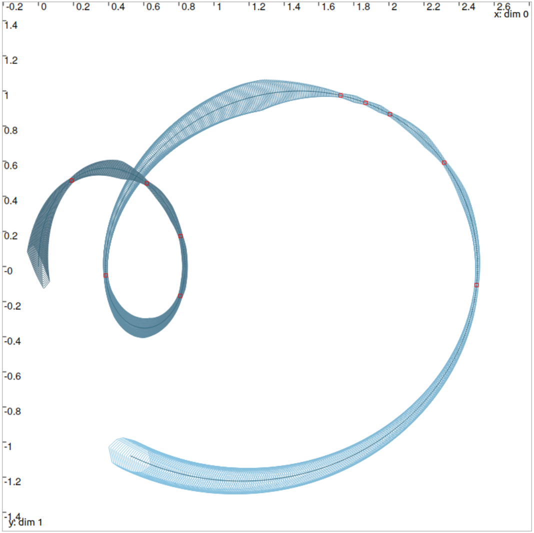


Forward computation
All observations
$\delta = 0.01$

Computation time: $\sim$2s

Test case

Resulting polygonal envelop



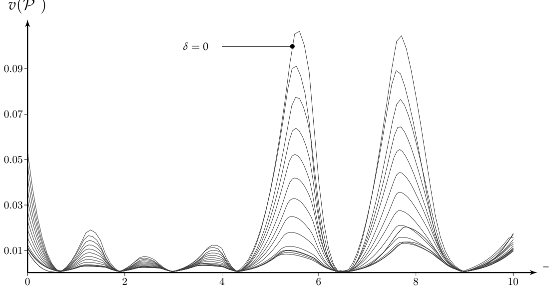
Forward/backward
All observations
$\delta = 0.01$

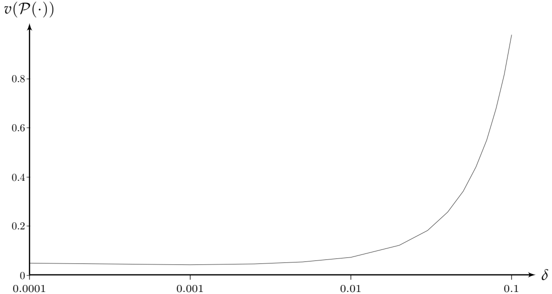
Computation time: $\sim$2s

Enclosure of $\mathbf{x}(0)$

Test case

Convergence to exact enclosure, with respect to δ





Section 5

Conclusion

Conclusion

Outline

New state estimator to approximate the state of a continuous-time linear system in a set-membership context, with a set of discrete measurements.

Conclusion

Outline

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 proposed method is:

- **accurate**:
  direct extension of the exact method used for discrete time systems,
- **guaranteed**:
  propagation of uncertainties made by interval analysis,
- **exact** if δ infinitely small,
  since it does not introduce any pessimism.

Conclusion

Limits

On dimensions:

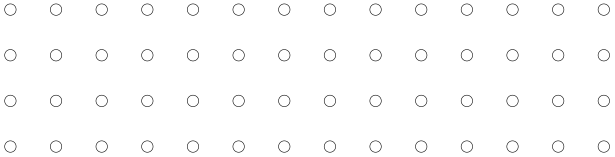
- polygonal implementation $\rightarrow$ difficult task, restricted to 2d cases at the moment

Conclusion

Limits

On dimensions:

- polygonal implementation $\rightarrow$ difficult task, restricted to 2d cases at the moment

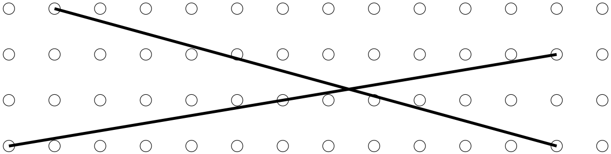


Conclusion

Limits

On dimensions:

- polygonal implementation $\rightarrow$ difficult task, restricted to 2d cases at the moment

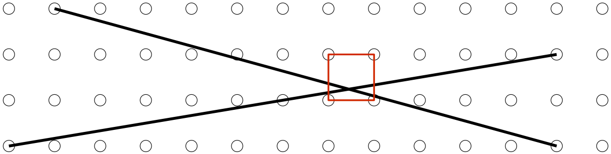


Conclusion

Limits

On dimensions:

- polygonal implementation → difficult task, restricted to 2d cases at the moment

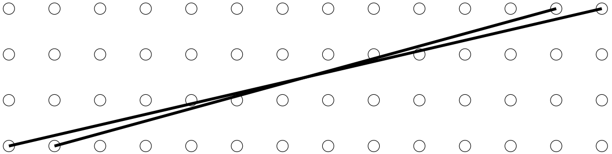


Conclusion

Limits

On dimensions:

- polygonal implementation $\rightarrow$ difficult task, restricted to 2d cases at the moment

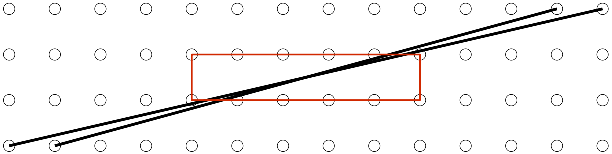


Conclusion

Limits

On dimensions:

- polygonal implementation $\rightarrow$ difficult task, restricted to 2d cases at the moment

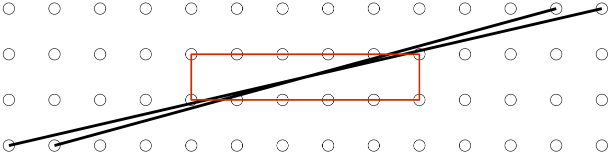


Conclusion

Limits

On dimensions:

- polygonal implementation $\rightarrow$ difficult task, restricted to 2d cases at the moment



On computation time:

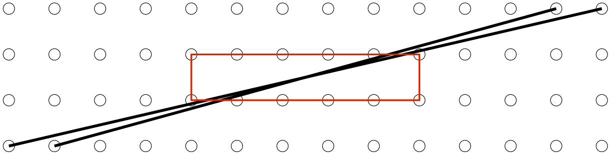
- number of vertices: simplification of polygons.

Conclusion

Limits

On dimensions:

- polygonal implementation $\rightarrow$ difficult task, restricted to 2d cases at the moment



On computation time:

- number of vertices: simplification of polygons.

On systems:

- the method cannot be extended directly to non-linear systems,
- the system must be time-invariant (computation of $e^{\mathbf{A}t}$).

Conclusion

Alternative to polygons: ellipsoids

Collaboration with Andreas Rauh.

- **An ellipsoidal predictor-corrector state estimation scheme for linear continuous-time systems with bounded parameters and bounded measurement errors**
  A. Rauh, S. Rohou, L. Jaulin, *Frontiers In Control Engineering*, 2022

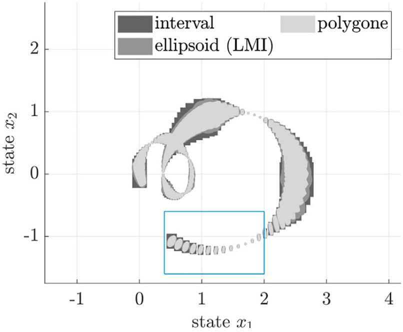
Conclusion

Alternative to polygons: ellipsoids

Collaboration with Andreas Rauh.

■ **An ellipsoidal predictor-corrector state estimation scheme for linear continuous-time systems with bounded parameters and bounded measurement errors**

A. Rauh, S. Rohou, L. Jaulin, *Frontiers In Control Engineering*, 2022



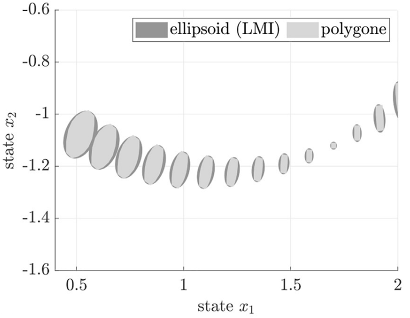
Conclusion

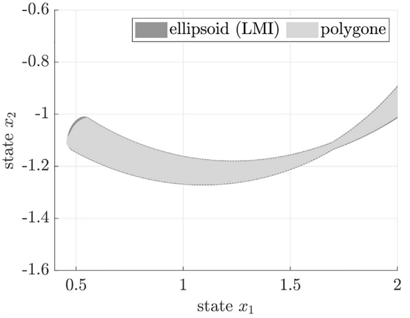
Alternative to polygons: ellipsoids

Collaboration with Andreas Rauh.

An ellipsoidal predictor-corrector state estimation scheme for linear continuous-time systems with bounded parameters and bounded measurement errors

A. Rauh, S. Rohou, L. Jaulin, *Frontiers In Control Engineering*, 2022





Conclusion

Code example – Codac library – (C++, Python available)

```
/* =========== PROBLEM DEFINITION =========== */

double dt = 0.01;
auto tdomain = create_tdomain(Interval(0,10), dt, true);

TFunction f_u("cos(t)");

Matrix A({{0,1},{-1,-1}});
Vector b({0,1});

//
```

Conclusion

Code example – Codac library – (C++, Python available)

```
/* =========== PROBLEM DEFINITION =========== */

double dt = 0.01;
auto tdomain = create_tdomain(Interval(0,10), dt, true);

TFunction f_u("cos(t)");

Matrix A({{0,1},{-1,-1}});
Vector b({0,1});

/* =========== SIMULATING THE TRUTH =========== */

TrajectoryVector x_truth(tdomain->t0_tf(), TFunction("( \
  sin(t) - (2*exp(-t/2)*sin((sqrt(3)*t)/2))/sqrt(3) ; \
  exp(-t/2)*sin(sqrt(3)*t/2)/sqrt(3)-sin(t)/2+sin(t)/2+cos(t)-exp(-t/2)*cos(sqrt(3)*t/2)\
  )"), dt/10.);

//
```

Conclusion

Code example – Codac library – (C++, Python available)

```
/* =========== PROBLEM DEFINITION =========== */

double dt = 0.01;
auto tdomain = create_tdomain(Interval(0,10), dt, true);

TFunction f_u("cos(t)");

Matrix A({{0,1},{-1,-1}});
Vector b({0,1});

/* =========== SIMULATING THE TRUTH =========== */

TrajectoryVector x_truth(tdomain->t0_tf(), TFunction("( \
  sin(t) - (2*exp(-t/2)*sin((sqrt(3)*t)/2))/sqrt(3) ; \
  exp(-t/2)*sin(sqrt(3)*t/2)/sqrt(3)-sin(t)/2+sin(t)/2+cos(t)-exp(-t/2)*cos(sqrt(3)*t/2)\
  )"), dt/10.);

/* =========== CREATING TUBES =========== */

Tube<ConvexPolygon> x(tdomain);
x.set(ConvexPolygon(Vector({0.,0.})), 0.); // setting initial condition
Tube<Interval> u(tdomain, f_u);
u.inflate(0.1); // u* + [-0.1,0.1]




//
```

Conclusion

Code example – Codac library – (C++, Python available)

```
/* =========== PROBLEM DEFINITION =========== */

double dt = 0.01;
auto tdomain = create_tdomain(Interval(0,10), dt, true);

TFunction f_u("cos(t)");

Matrix A({{0,1},{-1,-1}});
Vector b({0,1});

/* =========== SIMULATING THE TRUTH =========== */

TrajectoryVector x_truth(tdomain->t0_tf(), TFunction("( \
  sin(t) - (2*exp(-t/2)*sin((sqrt(3)*t)/2))/sqrt(3) ; \
  exp(-t/2)*sin(sqrt(3)*t/2)/sqrt(3)-sin(t)/2+sin(t)/2+cos(t)-exp(-t/2)*cos(sqrt(3)*t/2)\
  )"), dt/10.);

/* =========== CREATING TUBES =========== */

Tube<ConvexPolygon> x(tdomain);
x.set(ConvexPolygon(Vector({0.,0.})), 0.); // setting initial condition
Tube<Interval> u(tdomain, f_u);
u.inflate(0.1); // u* + [-0.1,0.1]

/* =========== CONTRACTING THE POLYGON TUBE =========== */

CtcLinobs ctc_linobs(A, b);
ctc_linobs.contract(x, u, TimePropag::FORWARD | TimePropag::BACKWARD);
//
```

Conclusion

Exact bounded-error continuous-time linear state estimator
S. Rohou, L. Jaulin, *Systems & Control Letters*, 2021

Questions?