Brunovsky decomposition for dynamic interval localization

Simon Rohou, Luc Jaulin

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

SWIM 2023, Angers
27^{th} June 2023

Section 1

Introduction

Introduction

Problem statement

Let us consider the following non-linear system:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)), & \text{(1a)} \\ y_i &= g(\mathbf{x}(t_i)), & \text{(1b)} \end{aligned}$$

- no prior knowledge about the states $\mathbf{x}(t) \in \mathbb{R}^n$
- but a discrete set of non-linear state observations $y_i \in \mathbb{R}$

Introduction

Problem statement

Let us consider the following non-linear system:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), & \text{(1a)} \\ y_i &= g(\mathbf{x}(t_i)), & \text{(1b)} \end{aligned}$$

- no prior knowledge about the states $\mathbf{x}(t) \in \mathbb{R}^n$
- but a discrete set of non-linear state observations $y_i \in \mathbb{R}$

The problem is difficult:

- non-linearities in $\mathbf{f}$, g
- uncertainties on $\mathbf{u}(\cdot)$, y_i, t_i, ...
- no initial condition

  $\Longrightarrow$ $\mathbf{x}(t) \in [-\infty, \infty]^n$

  $\Longrightarrow$ no linearization point

Introduction

Problem statement

Let us consider the following non-linear system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (1a)$$

$$y_i = g(\mathbf{x}(t_i)), \quad (1b)$$

- no prior knowledge about the states $\mathbf{x}(t) \in \mathbb{R}^n$
- but a discrete set of non-linear state observations $y_i \in \mathbb{R}$

The problem is difficult:

- non-linearities in $\mathbf{f}$, g
- uncertainties on $\mathbf{u}(\cdot)$, y_i, t_i, $\dots$
- no initial condition
  $\Longrightarrow$ $\mathbf{x}(t) \in [-\infty, \infty]^n$
  $\Longrightarrow$ no linearization point

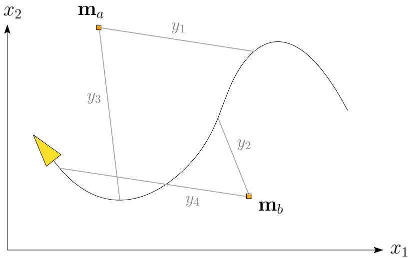
$\Rightarrow$ can be easily dealt with interval methods, but not in any cases

Introduction

Test case: full robotic state estimation

Landmarks-based localization of a mobile robot:

- $\mathbf{x} \in \mathbb{R}^4$
  - x_1, x_2: position
  - x_3: heading
  - x_4: speed
- discrete set of range-only measurements
  - $y_i \in \mathbb{R}$
  - measurements from known landmarks $\mathbf{m}_a$, $\mathbf{m}_b$



Introduction

Evolution equation $\mathbf{f}$ (for a wheeled robot)

Let us consider the system described by the following equations:

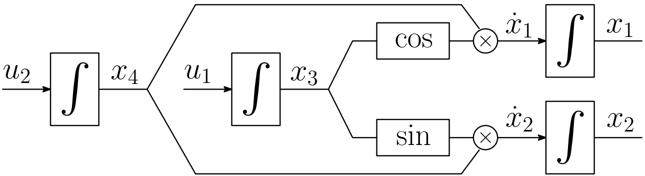
$$\mathbf{f}\left\{\begin{array}{lcl} \dot{x}_1 & = & x_4\cos(x_3) \\ \dot{x}_2 & = & x_4\sin(x_3) \\ \dot{x}_3 & = & u_1 \\ \dot{x}_4 & = & u_2 \end{array}\right. \tag{2}$$

Introduction

Evolution equation $\mathbf{f}$ (for a wheeled robot)

Let us consider the system described by the following equations:

$$\mathbf{f}\begin{cases} \dot{x}_1 &= x_4\cos(x_3) \\ \dot{x}_2 &= x_4\sin(x_3) \\ \dot{x}_3 &= u_1 \\ \dot{x}_4 &= u_2 \end{cases} \quad (2)$$

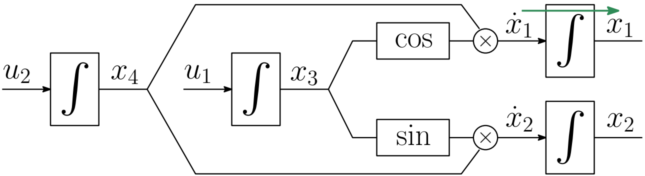


Introduction

Evolution equation $\mathbf{f}$ (for a wheeled robot)

Let us consider the system described by the following equations:

$$\mathbf{f}\left\{\begin{array}{lcl}\dot{x}_1 & = & x_4\cos(x_3)\\ \dot{x}_2 & = & x_4\sin(x_3)\\ \dot{x}_3 & = & u_1\\ \dot{x}_4 & = & u_2\end{array}\right. \tag{2}$$



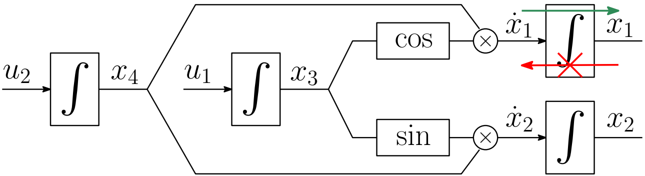
The problem is difficult when x_3 is unknown.
Information only comes from x_1, x_2, $\mathbf{u}$.

Introduction

Evolution equation $\mathbf{f}$ (for a wheeled robot)

Let us consider the system described by the following equations:

$$\mathbf{f}\begin{cases}\dot{x}_1 & = & x_4\cos(x_3)\\ \dot{x}_2 & = & x_4\sin(x_3)\\ \dot{x}_3 & = & u_1\\ \dot{x}_4 & = & u_2\end{cases} \tag{2}$$

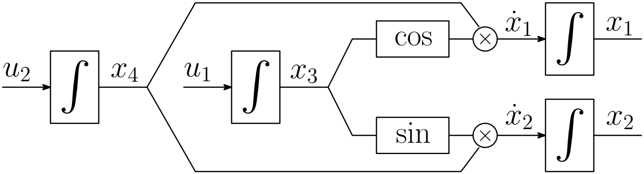


The problem is difficult when x_3 is unknown.
Information only comes from x_1, x_2, $\mathbf{u}$.

Introduction

Evolution equation $\mathbf{f}$ (for a wheeled robot)

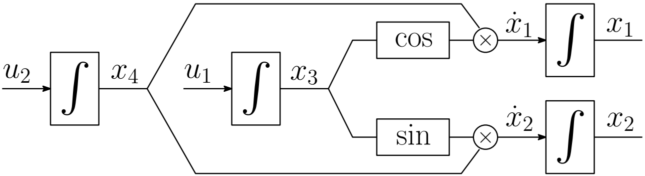
Overview of the «Brunovsky» approach:



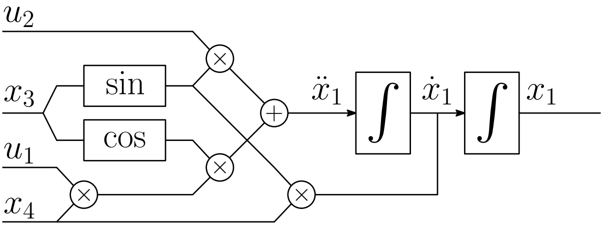
Introduction

Evolution equation $\mathbf{f}$ (for a wheeled robot)

Overview of the «Brunovsky» approach:



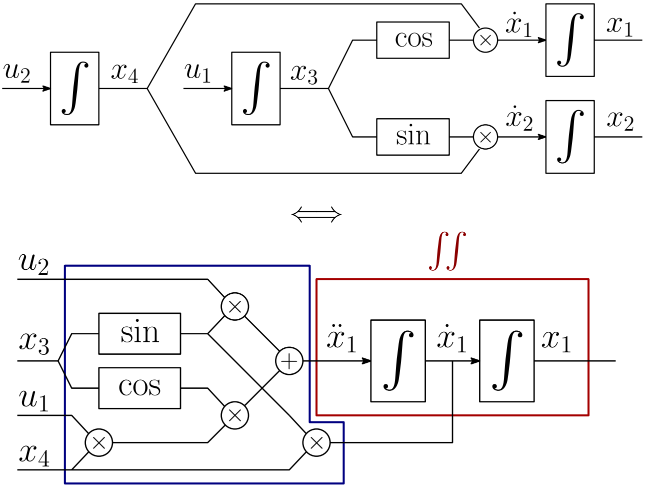
$$\Longleftrightarrow$$



Introduction

Evolution equation $\mathbf{f}$ (for a wheeled robot)

Overview of the «Brunovsky» approach:



Section 2

Brunovsky decomposition

Brunovsky decomposition

Flat systems

We consider the following system:

$$\left\{ \begin{array}{rcl} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{z} & = & \mathbf{h}(\mathbf{x}), \end{array} \right. \quad (3)$$

with $\mathbf{z} \in \mathbb{R}^m$: output vector used with a control point of view, and both $\mathbf{f}$ and $\mathbf{h}$ assumed to be smooth.
$\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{z} \in \mathbb{R}^m$.

The system is said to be flat if there exists two continuous functions ϕ and ψ and integers $\kappa_1, \dots, \kappa_m$ such that

$$\left\{ \begin{array}{rcl} \mathbf{x} & = & \phi\left(z_1, \dot{z}_1, \dots, z_1^{(\kappa_1-1)}, \dots, z_m, \dot{z}_m, \dots, z_m^{(\kappa_m-1)}\right) \\ \mathbf{u} & = & \psi\left(z_1, \dot{z}_1, \dots, z_1^{(\kappa_1)}, \dots, z_m, \dot{z}_m, \dots, z_m^{(\kappa_m)}\right). \end{array} \right. \quad (4)$$

Brunovsky decomposition

Flat systems

Usually, functions ϕ and ψ are obtained in two steps:

1. The **derivation step**, that computes symbolically $z_1, \dot{z}_1, \dots, z_1^{(\kappa_1)}, \dots\dots, z_m, \dot{z}_m, \dots, z_m^{(\kappa_m)}$ as functions of $\mathbf{x}$ and $\mathbf{u}$, using Eq. (3).
   We obtain an expression of the form

$$\begin{pmatrix} z_1 \\ \dot{z}_1 \\ \vdots \\ z_m^{(\kappa_m)} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}. \quad (5)$$

2. The **inversion step** in order to obtain ϕ and ψ

Brunovsky decomposition

Flat systems: example

Consider the system

$$\begin{cases} \dot{x}_1 & = & x_1 + x_2 \\ \dot{x}_2 & = & x_2^2 + u \\ z & = & x_1. \end{cases} \tag{6}$$

Brunovsky decomposition

Flat systems: example

Consider the system

$$\left\{\begin{array}{rcl}\dot{x}_1 & = & x_1 + x_2 \\ \dot{x}_2 & = & x_2^2 + u \\ z & = & x_1.\end{array}\right. \quad (6)$$

For the derivation step, we compute $z, \dot{z}, \ddot{z}, \ldots$ with respect to $\mathbf{x}$ and u until u occurs. We get

$$\lambda\left\{\begin{array}{rcl}z & = & x_1 \\ \dot{z} & = & \dot{x}_1 \ = \ x_1 + x_2 \\ \ddot{z} & = & \dot{x}_1 + \dot{x}_2 \ = \ x_1 + x_2 + x_2^2 + u.\end{array}\right. \quad (7)$$

Brunovsky decomposition

Flat systems: example

Consider the system

$$\left\{\begin{array}{rcl}\dot{x}_1 & = & x_1 + x_2\\ \dot{x}_2 & = & x_2^2 + u\\ z & = & x_1.\end{array}\right. \tag{6}$$

For the derivation step, we compute $z, \dot{z}, \ddot{z}, \ldots$ with respect to $\mathbf{x}$ and u until u occurs. We get

$$\lambda\left\{\begin{array}{rcl}z & = & x_1\\ \dot{z} & = & \dot{x}_1 \ = \ x_1 + x_2\\ \ddot{z} & = & \dot{x}_1 + \dot{x}_2 \ = \ x_1 + x_2 + x_2^2 + u.\end{array}\right. \tag{7}$$

Since we had to derive twice, we conclude that the Kronecker index is $\kappa = 2$ which corresponds to the dimension of $\mathbf{x} = (x_1, x_2)^\intercal$. As a consequence, the output z is flat.

Brunovsky decomposition

Flat systems: Brunovsky decomposition

The differential flat system:

- $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$
- with flat outputs $z_1, \ldots, z_m$
- and sensor outputs $\mathbf{y}$

Brunovsky decomposition

Flat systems: Brunovsky decomposition

The differential flat system:
- $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$
- with flat outputs $z_1, \dots, z_m$
- and sensor outputs $\mathbf{y}$

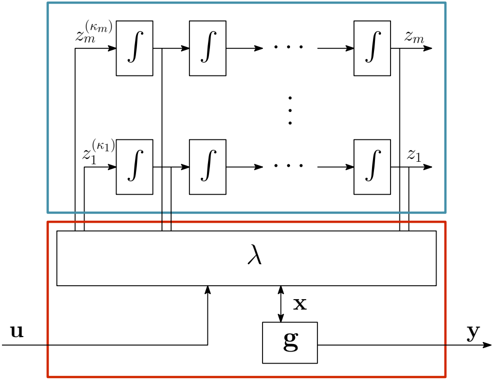
admits the following Brunovsky decomposition:

$$\left.\begin{array}{r} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{z} = \mathbf{h}(\mathbf{x}) \\ \mathbf{y} = \mathbf{g}(\mathbf{x}) \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{c} \begin{pmatrix} z_1 \\ \dot{z}_1 \\ \vdots \\ z_m^{(\kappa_m)} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix} \\ z_1^{(\kappa_1)} \overset{\int}{\rightarrow} \dots \overset{\int}{\rightarrow} \dot{z}_1 \overset{\int}{\rightarrow} z_1 \\ \vdots \\ z_m^{(\kappa_m)} \overset{\int}{\rightarrow} \dots \overset{\int}{\rightarrow} \dot{z}_m \overset{\int}{\rightarrow} z_m \\ \mathbf{y} = \mathbf{g}(\mathbf{x}) \end{array}\right. \qquad (8)$$

Brunovsky decomposition

Rewriting system

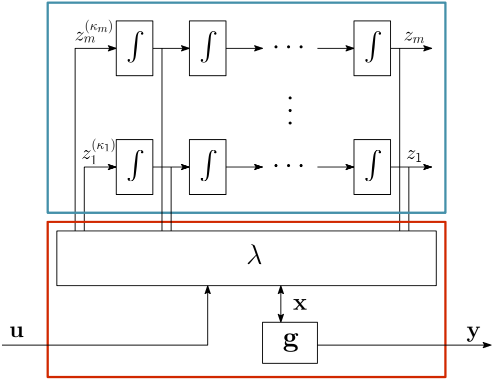
$$\left.\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t))\\ \mathbf{z} &= \mathbf{h}(\mathbf{x})\\ \mathbf{y} &= \mathbf{g}(\mathbf{x})\end{aligned}\right\} \Longleftrightarrow$$



Brunovsky decomposition

Rewriting system

$$\left.\begin{aligned}\dot{\mathbf{x}}(t)=\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t))\\ \mathbf{z}=\mathbf{h}(\mathbf{x})\\ \mathbf{y}=\mathbf{g}(\mathbf{x})\end{aligned}\right\}\iff$$

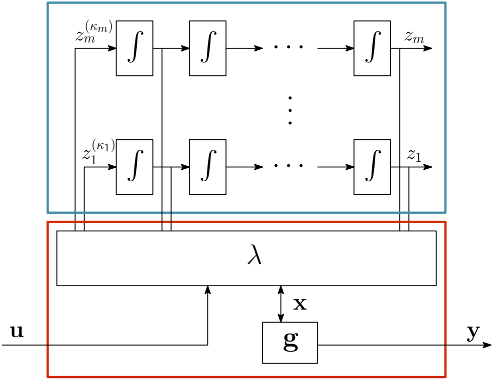


– Introducing so-called *Chains of integrators*: $\int \to \int \to \int \to \dots$

Brunovsky decomposition

Rewriting system

$$\left.\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t))\\ \mathbf{z} &= \mathbf{h}(\mathbf{x})\\ \mathbf{y} &= \mathbf{g}(\mathbf{x})\end{aligned}\right\}\Longleftrightarrow$$



- Introducing so-called *Chains of integrators*: $\int \rightarrow \int \rightarrow \int \rightarrow \dots$
- Integrator operations $\int$ are separated from non-linear relations: λ, $\mathbf{g}$

Brunovsky decomposition

Rewriting system: application on the wheeled robot

Decomposition of the evolution function $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$:

$$\left.\begin{array}{rl} (i) & \dot{x}_1 = x_4 \cos(x_3) \\ (ii) & \dot{x}_2 = x_4 \sin(x_3) \\ (iii) & \dot{x}_3 = u_1 \\ (iv) & \dot{x}_4 = u_2 \end{array}\right\}$$

Brunovsky decomposition

Rewriting system: application on the wheeled robot

Decomposition of the evolution function $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$:
From the literature, let us take $\mathbf{z} = (x_1, x_2)^\intercal$

$$\left.\begin{array}{rl} (i) & \dot{x}_1 = x_4\cos(x_3) \\ (ii) & \dot{x}_2 = x_4\sin(x_3) \\ (iii) & \dot{x}_3 = u_1 \\ (iv) & \dot{x}_4 = u_2 \end{array}\right\} \iff \begin{array}{ll} (I) & \left\{ \begin{pmatrix} z_1 \\ z_2 \\ \\ \\ \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \\ \\ \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \right. \\ (II) & \left\{ \begin{array}{l} z_1 \\ z_2 \end{array} \right. \end{array}$$

Brunovsky decomposition

Rewriting system: application on the wheeled robot

Decomposition of the evolution function $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$:
From the literature, let us take $\mathbf{z} = (x_1, x_2)^\intercal$

$$\left.\begin{array}{rl} (i) & \dot{x}_1 = x_4 \cos(x_3) \\ (ii) & \dot{x}_2 = x_4 \sin(x_3) \\ (iii) & \dot{x}_3 = u_1 \\ (iv) & \dot{x}_4 = u_2 \end{array}\right\} \iff \begin{array}{ll} (I) & \left\{ \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \\ \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4 \cos(x_3) \\ x_4 \sin(x_3) \\ \\ \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \right. \\ (II) & \left\{ \begin{array}{l} \dot{z}_1 \xrightarrow{\int} z_1 \\ \dot{z}_2 \xrightarrow{\int} z_2 \end{array} \right. \end{array}$$

Brunovsky decomposition

Rewriting system: application on the wheeled robot

Decomposition of the evolution function $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$:
From the literature, let us take $\mathbf{z} = (x_1, x_2)^\intercal$

$$\left.\begin{array}{rl} (i) & \dot{x}_1 = x_4\cos(x_3) \\ (ii) & \dot{x}_2 = x_4\sin(x_3) \\ (iii) & \dot{x}_3 = u_1 \\ (iv) & \dot{x}_4 = u_2 \end{array}\right\} \iff (I) \left\{ \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4\cos(x_3) \\ x_4\sin(x_3) \\ u_2\cos(x_3) - u_1x_4\sin(x_3) \\ u_2\sin(x_3) + u_1x_4\cos(x_3) \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \right.$$

$$(II) \left\{ \begin{array}{l} \ddot{z}_1 \xrightarrow{\int} \dot{z}_1 \xrightarrow{\int} z_1 \\ \ddot{z}_2 \xrightarrow{\int} \dot{z}_2 \xrightarrow{\int} z_2 \end{array} \right.$$

Brunovsky decomposition

Rewriting system: application on the wheeled robot

Decomposition of the evolution function $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$:
From the literature, let us take $\mathbf{z} = (x_1, x_2)^\intercal$

$$\left.\begin{array}{rl} (i) & \dot{x}_1 = x_4 \cos(x_3) \\ (ii) & \dot{x}_2 = x_4 \sin(x_3) \\ (iii) & \dot{x}_3 = u_1 \\ (iv) & \dot{x}_4 = u_2 \end{array}\right\} \iff \begin{array}{ll} (I) & \left\{ \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4 \cos(x_3) \\ x_4 \sin(x_3) \\ u_2 \cos(x_3) - u_1 x_4 \sin(x_3) \\ u_2 \sin(x_3) + u_1 x_4 \cos(x_3) \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \right. \\ \\ (II) & \left\{ \begin{array}{l} \ddot{z}_1 \xrightarrow{\int} \dot{z}_1 \xrightarrow{\int} z_1 \\ \ddot{z}_2 \xrightarrow{\int} \dot{z}_2 \xrightarrow{\int} z_2 \end{array} \right. \end{array}$$

- Block (I) is only made of non-linear static equations
- Block (II) is made of pure chains of integrators

Brunovsky decomposition

Rewriting system: application on the wheeled robot

$$\left\{ \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4\cos(x_3) \\ x_4\sin(x_3) \\ u_2\cos(x_3) - u_1x_4\sin(x_3) \\ u_2\sin(x_3) + u_1x_4\cos(x_3) \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \right.$$

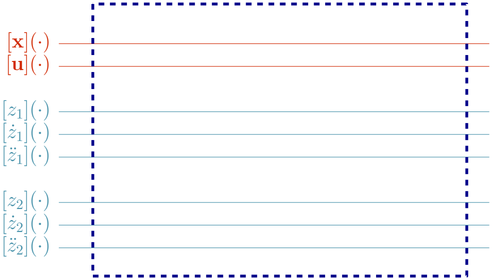
$$\left\{ \begin{array}{l} \ddot{z}_1 \xrightarrow{\int} \dot{z}_1 \xrightarrow{\int} z_1 \\ \ddot{z}_2 \xrightarrow{\int} \dot{z}_2 \xrightarrow{\int} z_2 \end{array} \right.$$

Brunovsky decomposition

Rewriting system: application on the wheeled robot

$$\left\{ \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4 \cos(x_3) \\ x_4 \sin(x_3) \\ u_2 \cos(x_3) - u_1 x_4 \sin(x_3) \\ u_2 \sin(x_3) + u_1 x_4 \cos(x_3) \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \right.$$

$$\left\{ \begin{array}{l} \ddot{z}_1 \xrightarrow{\int} \dot{z}_1 \xrightarrow{\int} z_1 \\ \ddot{z}_2 \xrightarrow{\int} \dot{z}_2 \xrightarrow{\int} z_2 \end{array} \right.$$

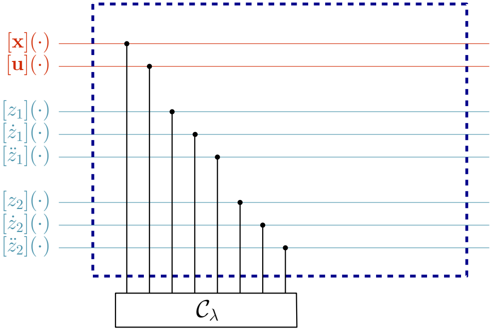


Brunovsky decomposition

Rewriting system: application on the wheeled robot

$$\left\{ \begin{array}{l} \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4 \cos(x_3) \\ x_4 \sin(x_3) \\ u_2 \cos(x_3) - u_1 x_4 \sin(x_3) \\ u_2 \sin(x_3) + u_1 x_4 \cos(x_3) \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \end{array} \right.$$

$$\left\{ \begin{array}{l} \ddot{z}_1 \overset{\int}{\rightarrow} \dot{z}_1 \overset{\int}{\rightarrow} z_1 \\ \ddot{z}_2 \overset{\int}{\rightarrow} \dot{z}_2 \overset{\int}{\rightarrow} z_2 \end{array} \right.$$

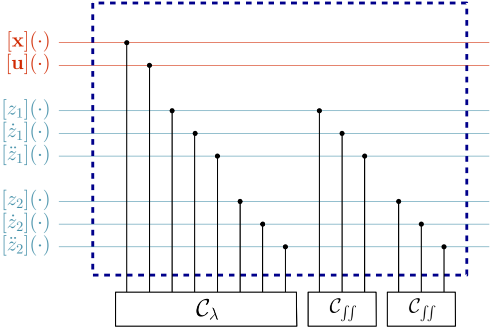


Brunovsky decomposition

Rewriting system: application on the wheeled robot

$$\left\{\begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4\cos(x_3) \\ x_4\sin(x_3) \\ u_2\cos(x_3) - u_1x_4\sin(x_3) \\ u_2\sin(x_3) + u_1x_4\cos(x_3) \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})}\right.$$

$$\left\{\begin{array}{l} \ddot{z}_1 \xrightarrow{\int} \dot{z}_1 \xrightarrow{\int} z_1 \\ \ddot{z}_2 \xrightarrow{\int} \dot{z}_2 \xrightarrow{\int} z_2 \end{array}\right.$$

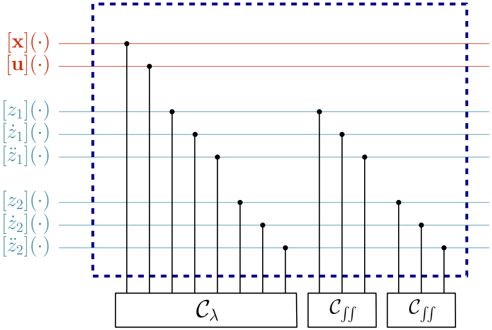


Brunovsky decomposition

Rewriting system: application on the wheeled robot

$$\left\{ \begin{pmatrix} z_1 \\ z_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \ddot{z}_1 \\ \ddot{z}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_4 \cos(x_3) \\ x_4 \sin(x_3) \\ u_2 \cos(x_3) - u_1 x_4 \sin(x_3) \\ u_2 \sin(x_3) + u_1 x_4 \cos(x_3) \end{pmatrix}}_{\lambda(\mathbf{x},\mathbf{u})} \right.$$

$$\left\{ \begin{array}{l} \ddot{z}_1 \xrightarrow{\int} \dot{z}_1 \xrightarrow{\int} z_1 \\ \ddot{z}_2 \xrightarrow{\int} \dot{z}_2 \xrightarrow{\int} z_2 \end{array} \right.$$



Involved contractors:

- $\mathcal{C}_\lambda([\mathbf{x}], [\mathbf{u}], [\mathbf{z}], [\dot{\mathbf{z}}], [\ddot{\mathbf{z}}])$
- $\mathcal{C}_{\int\int}([z_1](\cdot), [\dot{z}_1](\cdot), [\ddot{z}_1](\cdot))$
- $\mathcal{C}_{\int\int}([z_2](\cdot), [\dot{z}_2](\cdot), [\ddot{z}_2](\cdot))$

Section 3

The integrator chain contractor $\mathcal{C}_{\int\int}$

The integrator chain contractor $\mathcal{C}_{\int\int}$

Definition

A dedicated integrator chain contractor, denoted by $\mathcal{C}_{\int\int}$, has to be provided for:

$$z^{(\kappa)} \xrightarrow{\int} \dots \xrightarrow{\int} \dot{z} \xrightarrow{\int} z \tag{9}$$

$\longrightarrow$ it allows to accurately propagate information from one signal through its primitives and derivatives.

The integrator chain contractor $\mathcal{C}_{\int\int}$

Definition

A dedicated integrator chain contractor, denoted by $\mathcal{C}_{\int\int}$, has to be provided for:

$$z^{(\kappa)} \xrightarrow{\int} \dots \xrightarrow{\int} \dot{z} \xrightarrow{\int} z \tag{9}$$

$\longrightarrow$ it allows to accurately propagate information from one signal through its primitives and derivatives.

What about a decomposition?

$$z^{(\kappa)} \xrightarrow{\int} z^{(\kappa-1)},\ \dots\ ,\ddot{z} \xrightarrow{\int} \dot{z},\ \dots\ ,\dot{z} \xrightarrow{\int} z. \tag{10}$$

The integrator chain contractor $\mathcal{C}_{\int\int}$

Definition

A dedicated integrator chain contractor, denoted by $\mathcal{C}_{\int\int}$, has to be provided for:

$$z^{(\kappa)} \xrightarrow{\int} \dots \xrightarrow{\int} \dot{z} \xrightarrow{\int} z \tag{9}$$

$\longrightarrow$ it allows to accurately propagate information from one signal through its primitives and derivatives.

What about a decomposition?

$$z^{(\kappa)} \xrightarrow{\int} z^{(\kappa-1)}, \ \dots\ , \ddot{z} \xrightarrow{\int} \dot{z}, \ \dots\ , \dot{z} \xrightarrow{\int} z. \tag{10}$$

Strong wrapping effect

The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Let us consider the integrator chain constraint involving the signals $\left(z^{(0)}, z^{(1)}, \dots, z^{(\kappa)}, w\right)$ and defined as:

$$w \xrightarrow{\int} z^{(\kappa)} \xrightarrow{\int} \dots \xrightarrow{\int} \dot{z} \xrightarrow{\int} z \tag{11}$$

The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Let us consider the integrator chain constraint involving the signals $\left(z^{(0)}, z^{(1)}, \dots, z^{(\kappa)}, w\right)$ and defined as:

$$w \xrightarrow{\int} z^{(\kappa)} \xrightarrow{\int} \dots \xrightarrow{\int} \dot{z} \xrightarrow{\int} z \tag{11}$$

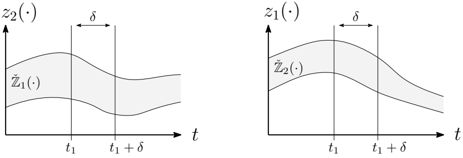
This chain can be cast into the following **linear system**:

$$\dot{\mathbf{z}}(t) = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \cdots & \\ 0 & 0 & 1 & & \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}}_{\mathbf{A}} \mathbf{z}(t) + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ \\ 1 \end{pmatrix}}_{\mathbf{B}} w(t), \tag{12}$$

where $w(\cdot)$ is known to be inside a tube $[w](\cdot)$.

The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator



Considering for instance the chain:

$w \xrightarrow{\int} z_2 \xrightarrow{\int} z_1$

and prior 2d sets $\check{\mathbb{Z}}(\cdot)$ implemented as tubes $[z_1] \times [z_2](\cdot)$ (upper part of the figure)

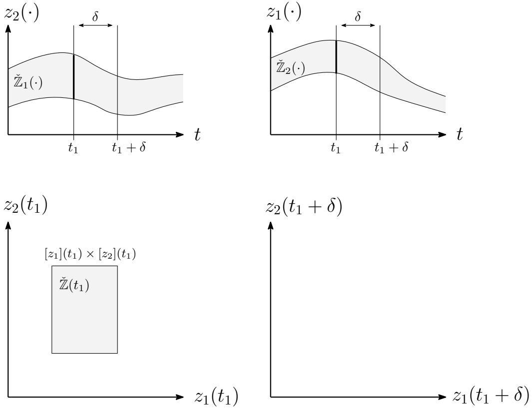
The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Considering for instance the chain:

$$w \xrightarrow{\int} z_2 \xrightarrow{\int} z_1$$

and prior 2d sets $\check{\mathbb{Z}}(\cdot)$ implemented as tubes $[z_1] \times [z_2](\cdot)$ (upper part of the figure)



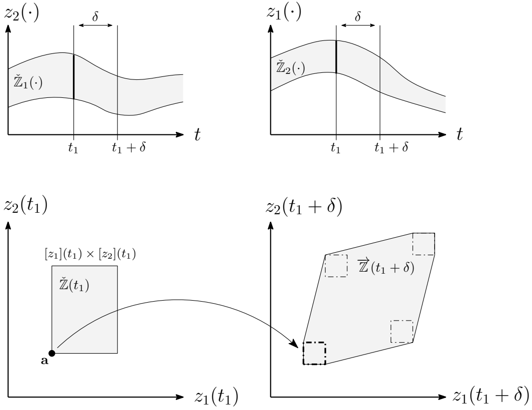
The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Considering for instance the chain:

$$w \xrightarrow{\int} z_2 \xrightarrow{\int} z_1$$

and prior 2d sets $\check{\mathbb{Z}}(\cdot)$ implemented as tubes $[z_1] \times [z_2](\cdot)$ (upper part of the figure)



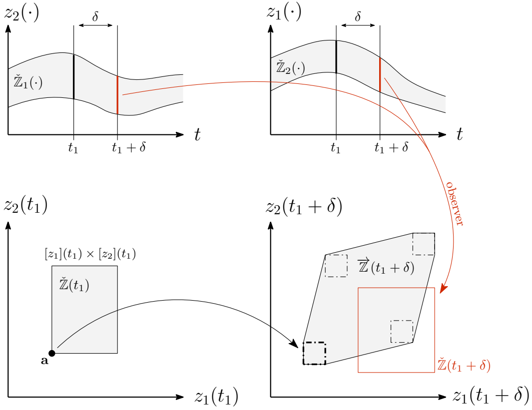
The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Considering for instance the chain:

$w \xrightarrow{\int} z_2 \xrightarrow{\int} z_1$

and prior 2d sets $\check{\mathbb{Z}}(\cdot)$ implemented as tubes $[z_1] \times [z_2](\cdot)$ (upper part of the figure)



State observations are processed as restrictions from the tubes.

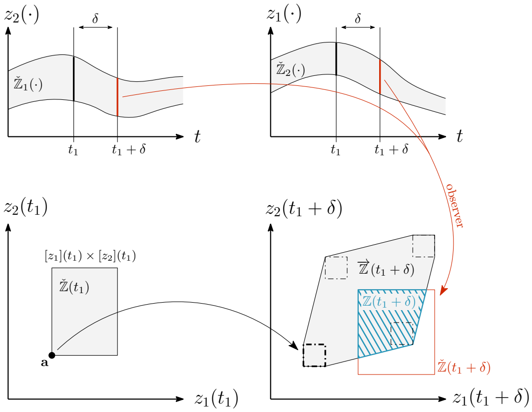
The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Considering for instance the chain:

$w \xrightarrow{\int} z_2 \xrightarrow{\int} z_1$

and prior 2d sets $\check{\mathbb{Z}}(\cdot)$ implemented as tubes $[z_1] \times [z_2](\cdot)$ (upper part of the figure)



One computation step of $\mathcal{C}_{\int\int}([z1](\cdot),[z2](\cdot),[w](\cdot)$
(result is the blue hatched part).

State observations are processed as restrictions from the tubes.

The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Minimal outputs can be obtained, as restrictions always correspond to the intersection of a polygon and a box, which can be computed accurately.

The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Minimal outputs can be obtained, as restrictions always correspond to the intersection of a polygon and a box, which can be computed accurately.

$\mathcal{C}_{\mathrm{linobs}}$: an optimal contractor for systems $\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{w}(t)$.

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

The integrator chain contractor $\mathcal{C}_{\int\int}$

Linear state estimator

Minimal outputs can be obtained, as restrictions always correspond to the intersection of a polygon and a box, which can be computed accurately.

$\mathcal{C}_{\text{linobs}}$: an optimal contractor for systems $\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{w}(t)$.

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

■ **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

Section 4

Back to the localization problem

Back to the localization problem

Contractor network

Mobile robotic state equations:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (13a)$$
$$y_i = g(\mathbf{x}(t_i)), \quad (13b)$$

Corresponding list of contractors,
resulting from the Brunovsky decomposition:

- $\mathcal{C}_\lambda([\mathbf{x}], [\mathbf{u}], [\mathbf{z}], [\dot{\mathbf{z}}], [\ddot{\mathbf{z}}])$
- $\mathcal{C}_{\int\int}([z_1](\cdot), [\dot{z_1}](\cdot), [\ddot{z_1}](\cdot))$
- $\mathcal{C}_{\int\int}([z_2](\cdot), [\dot{z_2}](\cdot), [\ddot{z_2}](\cdot))$
- $\mathcal{C}_g([\mathbf{x}](t_i), [y_i])$

Back to the localization problem

Reproducible example

Unknown states:

$$\mathbf{x}(t) = \begin{pmatrix} 10\cos(t) \\ 5\sin(2t) \\ \mathrm{atan2}\left(10\cos(2t), -10\sin(t)\right) \\ \sqrt{(-10\sin(t))^{2} + (10\cos(2t))^{2}} \end{pmatrix} \quad (14)$$

Known inputs:

$$\mathbf{u}(t) = \begin{pmatrix} \dfrac{2\sin(t)\sin(2t) + \cos(t)\cos(2t)}{\sin^2(t) + \cos^2(2t)} \\ \dfrac{10\cos(t)\cdot\sin(t) - 20\cos(2t)\cdot\sin(2t)}{\sqrt{\sin^2(t) + \cos^2(2t)}} \end{pmatrix} \quad (15)$$

Observation equation:

$$y^j(t_i) = \sqrt{\left(x_1(t_i) - m_1^j\right)^2 + \left(x_2(t_i) - m_2^j\right)^2} \quad (16)$$

Back to the localization problem

Reproducible example

Two known landmarks: $\mathbf{m}^a = (-5, 6)$ and $\mathbf{m}^b = (0, -4)$

t_i	$[y^a](t_i)$
0.75	[12.333,12.383]
2.25	[10.938,10.988]

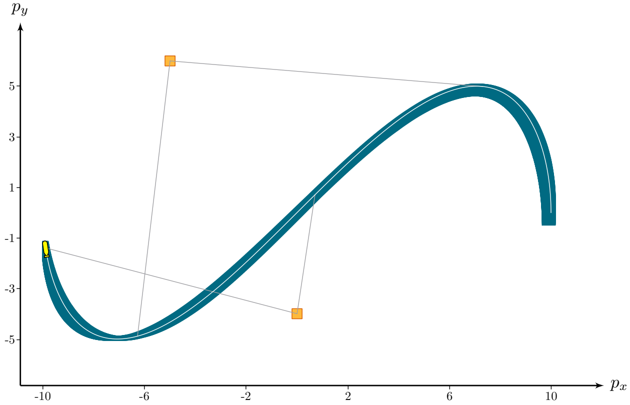
t_i	$[y^b](t_i)$
1.50	[4.733,4.783]
3.00	[10.211,10.261]

Table: Set of four bounded measurements $(t_i, [y](t_i))$.

The simulation is run for $t \in [0, 3]$.

Back to the localization problem

Results

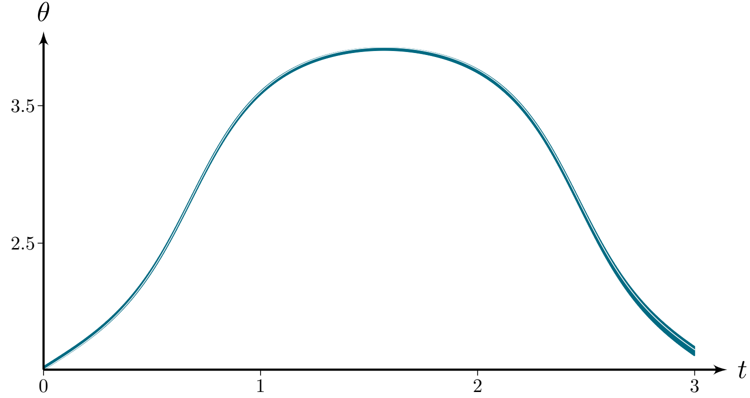


Set $[\mathbf{x}](\cdot)$ of feasible states projected in two dimensions. The unknown planar trajectory remains enclosed in the tube.

The simulation runs in 36 seconds.

Back to the localization problem

Results



The tube $[x_3](\cdot)$ of feasible headings. The actual but unknown truth is plotted in white and guaranteed to be enclosed in the computed $[x_3](\cdot)$.

The simulation runs in 36 seconds.

Section 5

Conclusion

Conclusion

Interval Brunovsky decomposition for non-linear systems

The problem is difficult due to:

- non-linearities in $\mathbf{f}$, g
- uncertainties on $\mathbf{u}(\cdot)$, y_i, t_i, $\dots$
- no initial condition on $\mathbf{x} \Longrightarrow$ no linearization point

Conclusion

Interval Brunovsky decomposition for non-linear systems

The problem is difficult due to:

- non-linearities in $\mathbf{f}$, g
- uncertainties on $\mathbf{u}(\cdot)$, y_i, t_i, $\dots$
- no initial condition on $\mathbf{x} \Longrightarrow$ no linearization point

Easily dealt with a **Brunovsky approach coupled with interval methods**, if the system is flat:

1. symbolic decomposition: the differential part is separated from the non-linear equations
2. an interval method is applied, with an optimal operator for the differential part expressed as a linear system

Conclusion

Brunovsky decomposition for dynamic interval localization
S. Rohou, L. Jaulin, *IEEE Transactions on Automatic Control*, 2023

Code available in the Codac library:
Interval tools for constraint programming over reals, trajectories and sets.
http://codac.io