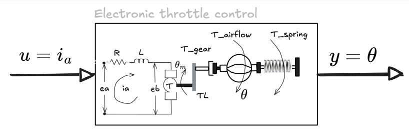
On Continuous-Time Sparse Identification of Nonlinear Polynomial Systems

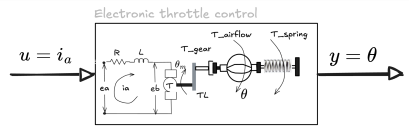
Mazen Alamir

CNRS-University of Grenoble-Alpes, France.

Problem statement



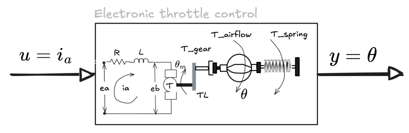
Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Problem statement



Identify a **SISO** model of the form:

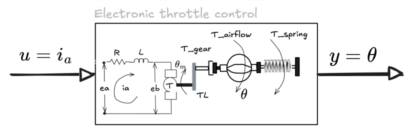
$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

NOTA: SISO assumption is not mandatory:

$$\forall\ i \in \{1, \dots, n_y\},$$

$$y_i^{(n_i)} = \mathcal{P}_i\Big(\underbrace{\big\{\{y_j^{(\kappa)}\}_{\kappa=0}^{n_j^{(i)}-1}\big\}_{j=1}^{n_y}, \{u_\ell\}_{\ell=1}^{n_u}}_{z_i}\Big)$$

Problem statement



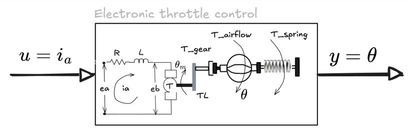
Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Problem statement



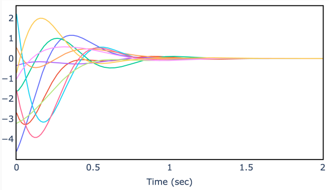
Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

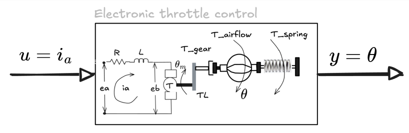
Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Typical short damped excitation instances



Problem statement



Identify a **SISO** model of the form:

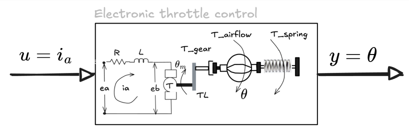
$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

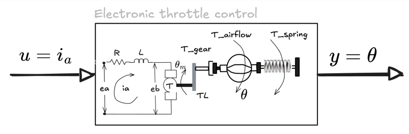
Use the resulting model in NMPC design.

Why Continuous-Time Identification?

- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.

Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

Why Continuous-Time Identification?

- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

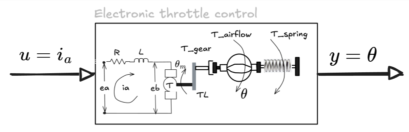
[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.

NOTA: SISO assumption is not mandatory:

$$\forall\ i \in \{1, \dots, n_y\},$$

$$y_i^{(n_i)} = \mathcal{P}_i\Big(\underbrace{\{\{y_j^{(\kappa)}\}_{\kappa=0}^{n_j^{(i)}-1}\}_{j=1}^{n_y}, \{u_\ell\}_{\ell=1}^{n_u}}_{z_i}\Big)$$

Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

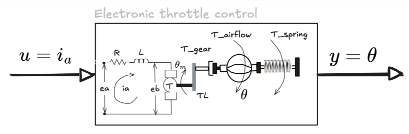
Why Continuous-Time Identification?

- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

✘ Needs high derivatives to be reconstructed!

[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.

Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

Why Continuous-Time Identification?

- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

✘ Needs high derivatives to be reconstructed!

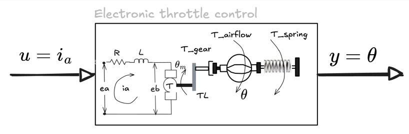
[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.

Automatica 180 (2025) 112469
Contents lists available at ScienceDirect
Automatica
journal homepage: www.elsevier.com/locate/automatica
ELSEVIER
Brief Paper

On reconstructing high derivatives of noisy time-series with confidence intervals☆

Mazen Alamir
Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France

Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

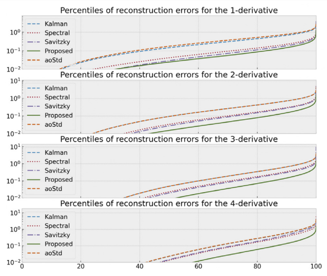
Use the resulting model in NMPC design.

Why Continuous-Time Identification?

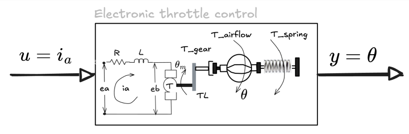
- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

✘ Needs high derivatives to be reconstructed!

[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.



Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

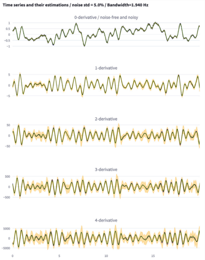
Use the resulting model in NMPC design.

Why Continuous-Time Identification?

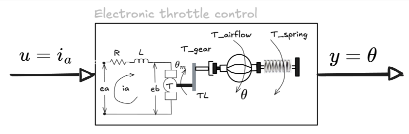
- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

✘ Needs high derivatives to be reconstructed!

[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.



Problem statement



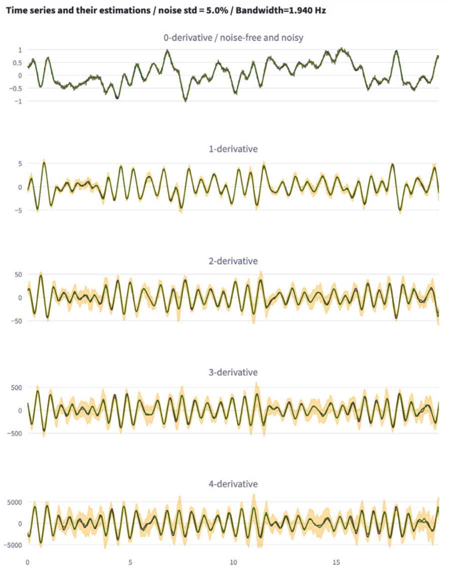
Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

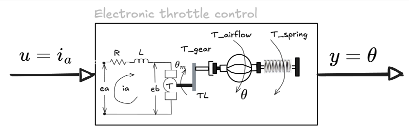
Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.



Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

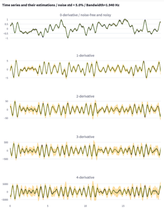
Use the resulting model in NMPC design.

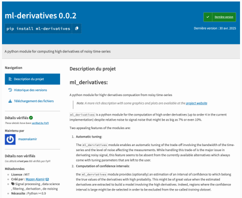
Why Continuous-Time Identification?

- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

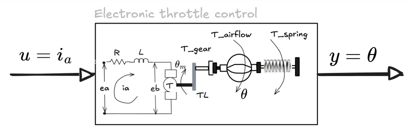
✘ Needs high derivatives to be reconstructed!

[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.





Problem statement



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

Why Continuous-Time Identification?

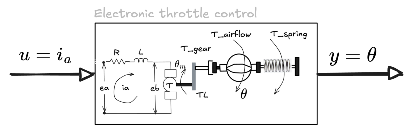
- ✓ Physically meaningful parameters
- ✓ Independence / sampling period
- ✓ Theoretically **less complicated** model[a]

✘ Needs high derivatives to be reconstructed!

[a] For $\dot{x} = f(x, u)$ with f polynomial of **degree** d, the RK4 sampled model is a polynomial of **degree** $4 \times d$.

```
pip install ml-derivatives
```

The need for parsimony



Identify a **SISO** model of the form:

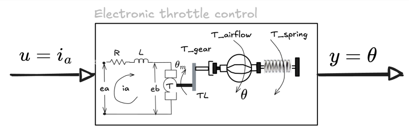
$$y^{(n)} = \mathcal{P}(y, \dot{y}, \ldots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

The need for parsimony



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

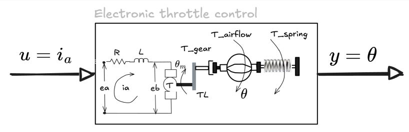
- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

What are the implication of the constraints?

The **training dataset** gathered from the short experiment **is only a tiny part** of the possible excursion of the arguments of $\mathcal{P}$.

The need for parsimony



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

What are the implication of the constraints?

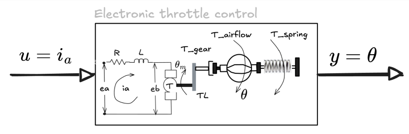
The **training dataset** gathered from the short experiment **is only a tiny part** of the possible excursion of the arguments of $\mathcal{P}$.

$\Rightarrow$ We need

- A model's **structure** and
- An identification **method**

that lead to **true generalization power**!

The need for parsimony



Identify a **SISO** model of the form:

$$y^{(n)} = \mathcal{P}(y, \dot{y}, \dots, y^{(n-1)}, u)$$

Under the following constraints:

- ✘ Unknown model / unknown n.
- ✘ Open-loop unstable dynamics
- ✘ **Only** short damped excitation allowed
- ✘ Noisy measurements

Use the resulting model in NMPC design.

What are the implication of the constraints?

The **training dataset** gathered from the short experiment **is only a tiny part** of the possible excursion of the arguments of $\mathcal{P}$.

$\Rightarrow$ We need

- A model's **structure** and
- An identification **method**

that lead to **true generalization power**!

$\Downarrow$

Parsimonious identification!

Problem

Identify a relationship:

$$y \approx P(x)$$

- $x \in \mathbb{R}^3$
- P: 4th order polynomial
- Fit **Train** models on $[-1, +1]^3$ / **Test** on $[-5, +5]^3$

Problem

Identify a relationship:

$$y \approx P(x)$$

- $x \in \mathbb{R}^3$
- P: 4th order polynomial
- Fit **Train** models on $[-1, +1]^3$ / **Test** on $[-5, +5]^3$

Alternatives

- Three DNN with $\neq$ complexities
- **Dense** `RidgeCV` $\circ$ `PolynomialFeatures`
- **plars** $\rightarrow$ **Sparse** polynomial solver

Problem

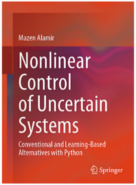
Identify a relationship:

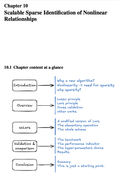
$$y \approx P(x)$$

- $x \in \mathbb{R}^3$
- P: 4th order polynomial
- Fit **Train** models on $[-1, +1]^3$ / **Test** on $[-5, +5]^3$

Alternatives

- Three DNN with $\neq$ complexities
- **Dense** `RidgeCV` ∘ `PolynomialFeatures`
- **plars** → **Sparse** polynomial solver





Chapter 10
Scalable Sparse Identification of Nonlinear Relationships

10.1 Chapter content at a glance

Problem

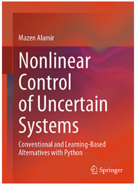
Identify a relationship:

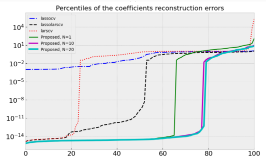
$$y \approx P(x)$$

- $x \in \mathbb{R}^3$
- P: 4th order polynomial
- Fit **Train** models on $[-1, +1]^3$ / **Test** on $[-5, +5]^3$

Alternatives

- Three DNN with $\neq$ complexities
- **Dense** `RidgeCV` ∘ `PolynomialFeatures`
- **plars** → **Sparse** polynomial solver





Problem

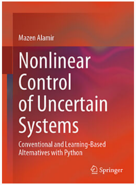
Identify a relationship:

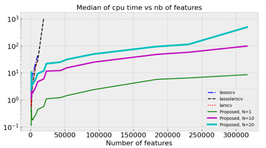
$$y \approx P(x)$$

- $x \in \mathbb{R}^3$
- P: 4th order polynomial
- Fit **Train** models on $[-1,+1]^3$ / **Test** on $[-5,+5]^3$

Alternatives

- Three DNN with $\neq$ complexities
- **Dense** `RidgeCV` $\circ$ `PolynomialFeatures`
- **plars** $\rightarrow$ **Sparse** polynomial solver



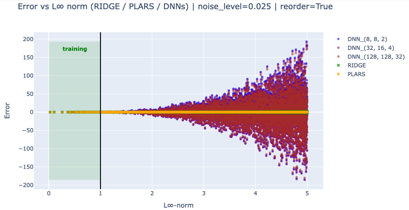


Problem

Identify a relationship:

$$y \approx P(x)$$

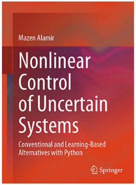
- $x \in \mathbb{R}^3$
- P: 4th order polynomial
- Fit **Train** models on $[-1, +1]^3$ / **Test** on $[-5, +5]^3$

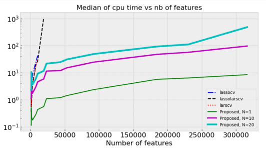


$$\texttt{error} = \frac{\texttt{percentile}(|y-\hat{y}|, 95)}{\texttt{median}(|y|)}$$

Alternatives

- Three DNN with ≠ complexities
- **Dense** `RidgeCV` ∘ `PolynomialFeatures`
- **plars** → **Sparse** polynomial solver





Problem

Identify a relationship:

$$y \approx P(x)$$

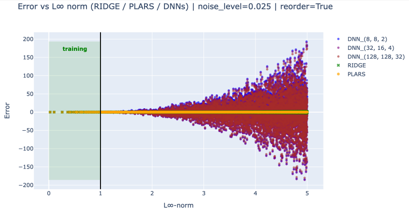
- $x \in \mathbb{R}^3$
- P: 4th order polynomial
- Fit **Train** models on $[-1,+1]^3$ / **Test** on $[-5,+5]^3$

Alternatives

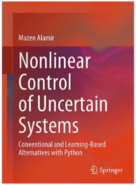
– Three DNN with $\neq$ complexities

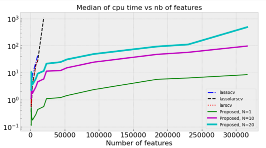
– **Dense** `RidgeCV` ∘ `PolynomialFeatures`

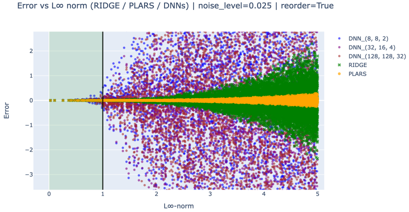
– **plars** → **Sparse** polynomial solver



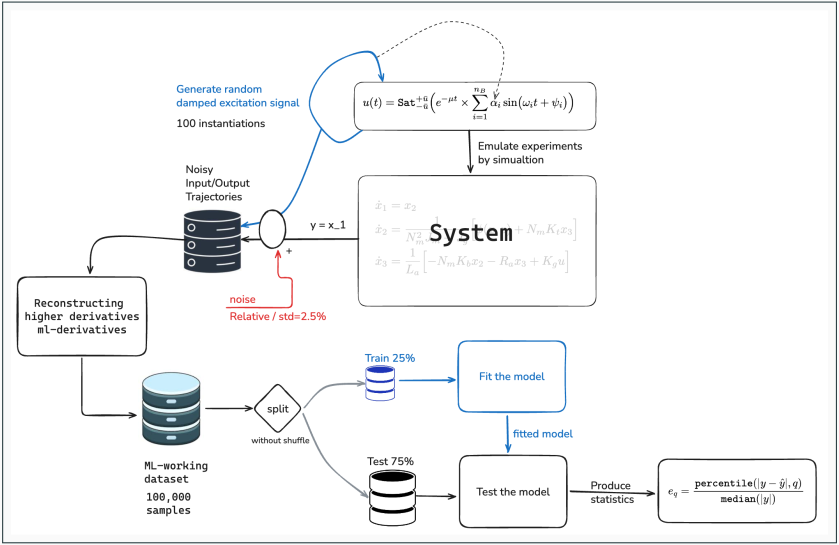
$$\texttt{error} = \frac{\texttt{percentile}(|y-\hat{y}|, 95)}{\texttt{median}(|y|)}$$





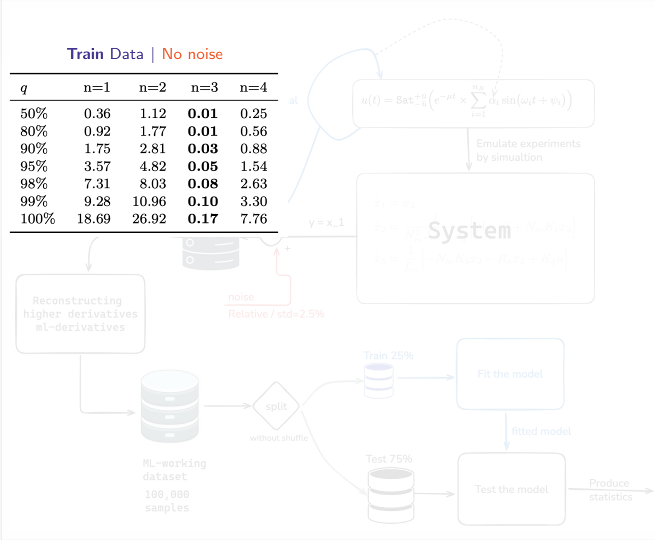


Data processing, models fitting & evaluation



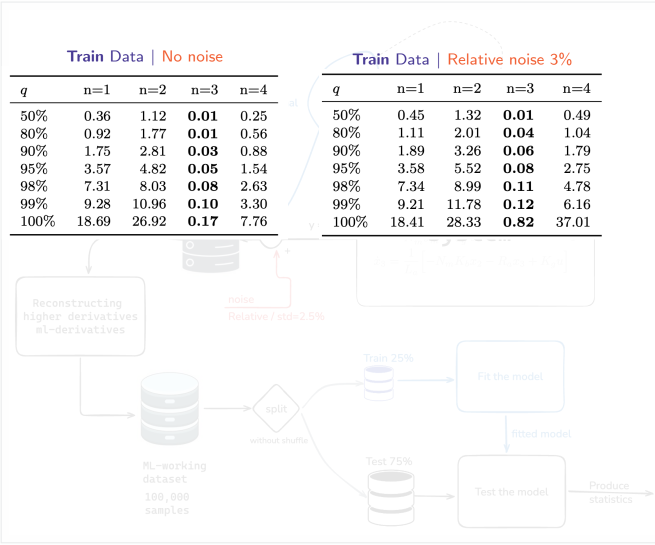
Train Data | No noise

q	n=1	n=2	n=3	n=4
50%	0.36	1.12	**0.01**	0.25
80%	0.92	1.77	**0.01**	0.56
90%	1.75	2.81	**0.03**	0.88
95%	3.57	4.82	**0.05**	1.54
98%	7.31	8.03	**0.08**	2.63
99%	9.28	10.96	**0.10**	3.30
100%	18.69	26.92	**0.17**	7.76



$$e_q = \frac{\texttt{percentile}(|y-\hat{y}|, q)}{\texttt{median}(|y|)}$$

Data processing, models fitting & evaluation



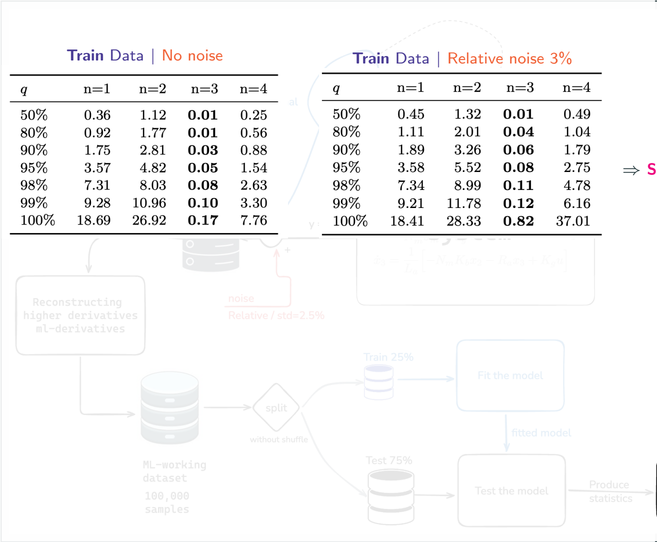
Train Data | No noise

q	n=1	n=2	n=3	n=4
50%	0.36	1.12	**0.01**	0.25
80%	0.92	1.77	**0.01**	0.56
90%	1.75	2.81	**0.03**	0.88
95%	3.57	4.82	**0.05**	1.54
98%	7.31	8.03	**0.08**	2.63
99%	9.28	10.96	**0.10**	3.30
100%	18.69	26.92	**0.17**	7.76

Train Data | Relative noise 3%

q	n=1	n=2	n=3	n=4
50%	0.45	1.32	**0.01**	0.49
80%	1.11	2.01	**0.04**	1.04
90%	1.89	3.26	**0.06**	1.79
95%	3.58	5.52	**0.08**	2.75
98%	7.34	8.99	**0.11**	4.78
99%	9.21	11.78	**0.12**	6.16
100%	18.41	28.33	**0.82**	37.01

$$e_q = \frac{\texttt{percentile}(|y-\hat{y}|, q)}{\texttt{median}(|y|)}$$



Train Data | No noise

q	n=1	n=2	n=3	n=4
50%	0.36	1.12	**0.01**	0.25
80%	0.92	1.77	**0.01**	0.56
90%	1.75	2.81	**0.03**	0.88
95%	3.57	4.82	**0.05**	1.54
98%	7.31	8.03	**0.08**	2.63
99%	9.28	10.96	**0.10**	3.30
100%	18.69	26.92	**0.17**	7.76

Train Data | Relative noise 3%

q	n=1	n=2	n=3	n=4
50%	0.45	1.32	**0.01**	0.49
80%	1.11	2.01	**0.04**	1.04
90%	1.89	3.26	**0.06**	1.79
95%	3.58	5.52	**0.08**	2.75
98%	7.34	8.99	**0.11**	4.78
99%	9.21	11.78	**0.12**	6.16
100%	18.41	28.33	**0.82**	37.01

$\Rightarrow$ **System's order $n = 3$.**

$$e_q = \frac{\texttt{percentile}(|y-\hat{y}|, q)}{\texttt{median}(|y|)}$$

Data processing, models fitting & evaluation

Train Data | No noise

q	n=1	n=2	n=3	n=4
50%	0.36	1.12	**0.01**	0.25
80%	0.92	1.77	**0.01**	0.56
90%	1.75	2.81	**0.03**	0.88
95%	3.57	4.82	**0.05**	1.54
98%	7.31	8.03	**0.08**	2.63
99%	9.28	10.96	**0.10**	3.30
100%	18.69	26.92	**0.17**	7.76

Train Data | Relative noise 3%

q	n=1	n=2	n=3	n=4
50%	0.45	1.32	**0.01**	0.49
80%	1.11	2.01	**0.04**	1.04
90%	1.89	3.26	**0.06**	1.79
95%	3.58	5.52	**0.08**	2.75
98%	7.34	8.99	**0.11**	4.78
99%	9.21	11.78	**0.12**	6.16
100%	18.41	28.33	**0.82**	37.01

$\Rightarrow$ **System's order $n = 3$.**

Test Data | Relative noise 3% | $d = 3$

q	50%	80%	90%	95%	98%	99%	100%
Error	**0.01**	**0.03**	**0.04**	**0.06**	**0.12**	**0.22**	**1.46**

$$e_q = \frac{\texttt{percentile}(|y-\hat{y}|, q)}{\texttt{median}(|y|)}$$

Data processing, models fitting & evaluation

Train Data | No noise

q	n=1	n=2	n=3	n=4
50%	0.36	1.12	**0.01**	0.25
80%	0.92	1.77	**0.01**	0.56
90%	1.75	2.81	**0.03**	0.88
95%	3.57	4.82	**0.05**	1.54
98%	7.31	8.03	**0.08**	2.63
99%	9.28	10.96	**0.10**	3.30
100%	18.69	26.92	**0.17**	7.76

Train Data | Relative noise 3%

q	n=1	n=2	n=3	n=4
50%	0.45	1.32	**0.01**	0.49
80%	1.11	2.01	**0.04**	1.04
90%	1.89	3.26	**0.06**	1.79
95%	3.58	5.52	**0.08**	2.75
98%	7.34	8.99	**0.11**	4.78
99%	9.21	11.78	**0.12**	6.16
100%	18.41	28.33	**0.82**	37.01

$\Rightarrow$ **System's order $n = 3$.**

Test Data | Relative noise 3% | $d = 3$

q	50%	80%	90%	95%	98%	99%	100%
Error	**0.01**	**0.03**	**0.04**	**0.06**	**0.12**	**0.22**	**1.46**

Test Data | Relative noise 3% | $d = 1$

q	50%	80%	90%	95%	98%	99%	100%
Error	0.03	0.07	0.09	0.13	0.24	0.36	1.82

$$e_q = \frac{\texttt{percentile}(|y-\hat{y}|, q)}{\texttt{median}(|y|)}$$

Train Data | No noise

q	n=1	n=2	n=3	n=4
50%	0.36	1.12	**0.01**	0.25
80%	0.92	1.77	**0.01**	0.56
90%	1.75	2.81	**0.03**	0.88
95%	3.57	4.82	**0.05**	1.54
98%	7.31	8.03	**0.08**	2.63
99%	9.28	10.96	**0.10**	3.30
100%	18.69	26.92	**0.17**	7.76

Train Data | Relative noise 3%

q	n=1	n=2	n=3	n=4
50%	0.45	1.32	**0.01**	0.49
80%	1.11	2.01	**0.04**	1.04
90%	1.89	3.26	**0.06**	1.79
95%	3.58	5.52	**0.08**	2.75
98%	7.34	8.99	**0.11**	4.78
99%	9.21	11.78	**0.12**	6.16
100%	18.41	28.33	**0.82**	37.01

$\Rightarrow$ **System's order $n = 3$.**

Test Data | Relative noise 3% | $d = 3$

q	50%	80%	90%	95%	98%	99%	100%
Error	**0.01**	**0.03**	**0.04**	**0.06**	**0.12**	**0.22**	**1.46**

Test Data | Relative noise 3% | $d = 1$

q	50%	80%	90%	95%	98%	99%	100%
Error	0.03	0.07	0.09	0.13	0.24	0.36	1.82

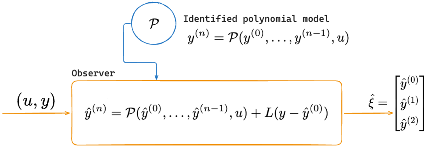
NOTA: When $d = 1$,

$y^{(3)}$ Affine in $y, \dot{y}, \ddot{y}$

But $\dot{y}$ and $\ddot{y}$ involve **nonlinear** relationships in x.

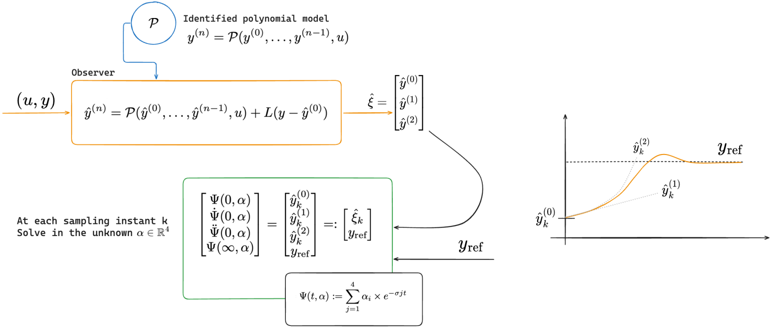
$$e_q = \frac{\texttt{percentile}(|y - \hat{y}|, q)}{\texttt{median}(|y|)}$$

NMPC Design

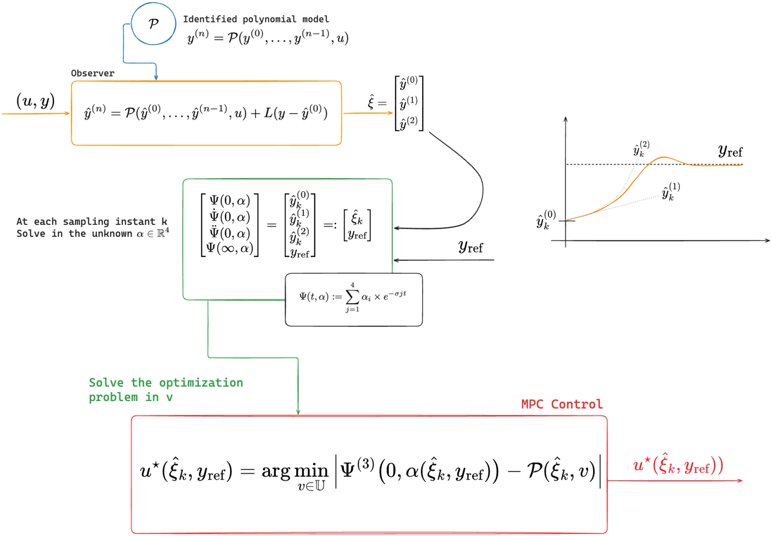


$\mathcal{P}$
Identified polynomial model
$y^{(n)} = \mathcal{P}(y^{(0)}, \dots, y^{(n-1)}, u)$
Observer
(u, y)
$\hat{y}^{(n)} = \mathcal{P}(\hat{y}^{(0)}, \dots, \hat{y}^{(n-1)}, u) + L(y - \hat{y}^{(0)})$
$\hat{\xi} = \begin{bmatrix} \hat{y}^{(0)} \\ \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix}$

NMPC Design


$\mathcal{P}$
Identified polynomial model
$y^{(n)} = \mathcal{P}(y^{(0)}, \dots, y^{(n-1)}, u)$
Observer
(u, y)
$\hat{y}^{(n)} = \mathcal{P}(\hat{y}^{(0)}, \dots, \hat{y}^{(n-1)}, u) + L(y - \hat{y}^{(0)})$
$\hat{\xi} = \begin{bmatrix} \hat{y}^{(0)} \\ \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix}$
At each sampling instant k
Solve in the unknown $\alpha \in \mathbb{R}^4$
$\begin{bmatrix} \Psi(0,\alpha) \\ \dot{\Psi}(0,\alpha) \\ \ddot{\Psi}(0,\alpha) \\ \Psi(\infty,\alpha) \end{bmatrix} = \begin{bmatrix} \hat{y}_k^{(0)} \\ \hat{y}_k^{(1)} \\ \hat{y}_k^{(2)} \\ y_{\mathrm{ref}} \end{bmatrix} =: \begin{bmatrix} \hat{\xi}_k \\ y_{\mathrm{ref}} \end{bmatrix}$
y_{ref}
$\Psi(t, \alpha) := \sum_{j=1}^{4} \alpha_i \times e^{-\sigma j t}$
$\hat{y}_k^{(2)}$
y_{ref}
$\hat{y}_k^{(1)}$
$\hat{y}_k^{(0)}$

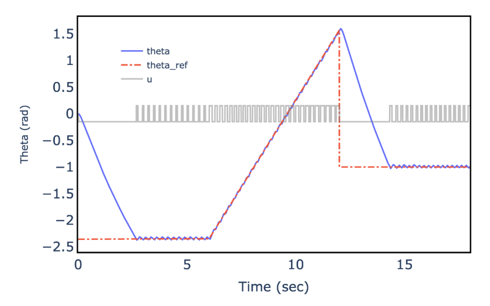
NMPC Design



$\mathcal{P}$
Identified polynomial model
$y^{(n)} = \mathcal{P}(y^{(0)}, \dots, y^{(n-1)}, u)$
Observer
(u, y)
$\hat{y}^{(n)} = \mathcal{P}(\hat{y}^{(0)}, \dots, \hat{y}^{(n-1)}, u) + L(y - \hat{y}^{(0)})$
$\hat{\xi} = \begin{bmatrix} \hat{y}^{(0)} \\ \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix}$
At each sampling instant k
Solve in the unknown $\alpha \in \mathbb{R}^4$
$\begin{bmatrix} \Psi(0,\alpha) \\ \dot{\Psi}(0,\alpha) \\ \ddot{\Psi}(0,\alpha) \\ \Psi(\infty,\alpha) \end{bmatrix} = \begin{bmatrix} \hat{y}_k^{(0)} \\ \hat{y}_k^{(1)} \\ \hat{y}_k^{(2)} \\ y_{\text{ref}} \end{bmatrix} =: \begin{bmatrix} \hat{\xi}_k \\ y_{\text{ref}} \end{bmatrix}$
y_{ref}
$\Psi(t,\alpha) := \sum_{j=1}^{4} \alpha_i \times e^{-\sigma j t}$
$\hat{y}_k^{(2)}$
y_{ref}
$\hat{y}_k^{(1)}$
$\hat{y}_k^{(0)}$
Solve the optimization problem in v
MPC Control
$u^\star(\hat{\xi}_k, y_{\text{ref}}) = \arg\min_{v \in \mathbb{U}} \left| \Psi^{(3)}(0, \alpha(\hat{\xi}_k, y_{\text{ref}})) - \mathcal{P}(\hat{\xi}_k, v) \right|$
$u^\star(\hat{\xi}_k, y_{\text{ref}}))$

Closed-loop under NMPC feedback

- ✓ Sampling period $\tau = 800\mu$sec
- ✓ Input blocked over 3 periods
- ✓ Control saturation $\bar{u} = 0.15$
- ✓ Real-time implementable.



Conclusion

Despite many advantages, **CT-identification was abandoned** because of the difficulty of reconstructing high derivatives from noisy measurement.

Conclusion

Despite many advantages, **CT-identification was abandoned** because of the difficulty of reconstructing high derivatives from noisy measurement.

Recent algorithmic advances (derivation, sparse polynomial identification) make CT-identification again a reasonable alternative.

Conclusion

Despite many advantages, **CT-identification was abandoned** because of the difficulty of reconstructing high derivatives from noisy measurement.

Recent algorithmic advances (derivation, sparse polynomial identification) make CT-identification again a reasonable alternative.

This paper provides a Proof-Of-Concept of using CT-polynomial **sparse** identification in identifying and controlling totally unknown, open-loop unstable system, **using a set of few safe & short damped excitations**.

Conclusion

Despite many advantages, **CT-identification was abandoned** because of the difficulty of reconstructing high derivatives from noisy measurement.

Recent algorithmic advances (derivation, sparse polynomial identification) make CT-identification again a reasonable alternative.

This paper provides a Proof-Of-Concept of using CT-polynomial **sparse** identification in identifying and controlling totally unknown, open-loop unstable system, **using a set of few safe & short damped excitations**.

Question?