On Explicit Partial Estimator Design for Nonlinear Systems with Parametric Uncertainties

Mazen Alamir

CNRS-University of Grenoble-Alpes, France.

Problem statement

Class of Systems

$$\dot{x} = f(x, u, p) \quad | \quad y = h(x, u, p)$$

$\rightarrow$ f, h known.

$\rightarrow$ p, uncertain | known statistics $\mathcal{P}$.

$\rightarrow$ y measurement vector **including u**

Problem statement

Class of Systems

$$\dot{x} = f(x, u, p) \quad | \quad y = h(x, u, p)$$

$\to$ f, h known.

$\to$ p, uncertain | known statistics $\mathcal{P}$.

$\to$ y measurement vector **including u**

Partial observation target

$$z = T(x, u, p)$$

$\to$ T known.

$\to$ Example: Control law: $z = k(x, p)$

MA. Partial Extended Observability Certification and Optimal Design of Moving Horizon Estimators. IEEE Transactions on Automatic Control, Volume 67, Issue 7, pages 3663-3669, July 2022.

Problem statement

Class of Systems

$$\dot{x} = f(x,u,p) \quad | \quad y = h(x,u,p)$$

$\rightarrow f, h$ known.

$\rightarrow p$, uncertain | known statistics $\mathcal{P}$.

$\rightarrow y$ measurement vector **including u**

Partial observation target

$$z = T(x,u,p)$$

$\rightarrow T$ known.

$\rightarrow$ Example: Control law: $z = k(x,p)$

MA. Partial Extended Observability Certification and Optimal Design of Moving Horizon Estimators. IEEE Transactions on Automatic Control, Volume 67, Issue 7, pages 3663-3669, July 2022.

Options

Extend the state $\rightarrow$ Ext-MHE $\rightarrow \hat{x}, \hat{p} \rightarrow \hat{z}$

CPU-expensive | full estimation | non observability

Problem statement

Class of Systems

$$\dot{x} = f(x,u,p) \quad | \quad y = h(x,u,p)$$

$\rightarrow f, h$ known.

$\rightarrow p$, uncertain | known statistics $\mathcal{P}$.

$\rightarrow y$ measurement vector **including u**

Partial observation target

$$z = T(x,u,p)$$

$\rightarrow T$ known.

$\rightarrow$ Example: Control law: $z = k(x,p)$

MA. Partial Extended Observability Certification and Optimal Design of Moving Horizon Estimators. IEEE Transactions on Automatic Control, Volume 67, Issue 7, pages 3663-3669, July 2022.

Options

Extend the state $\rightarrow$ Ext-MHE $\rightarrow \hat{x}, \hat{p} \rightarrow \hat{z}$
CPU-expensive | full estimation | non observability

Stochastic MHE $\rightarrow \hat{x}$:=argmin $\mathbb{E}_{\mathcal{P}}(\cdot)$
CPU-expensive | full estimation

Problem statement

Class of Systems

$$\dot{x} = f(x,u,p) \quad | \quad y = h(x,u,p)$$

$\rightarrow f, h$ known.
$\rightarrow p$, uncertain | known statistics $\mathcal{P}$.
$\rightarrow y$ measurement vector **including u**

Partial observation target

$$z = T(x,u,p)$$

$\rightarrow T$ known.
$\rightarrow$ Example: Control law: $z = k(x,p)$

MA. Partial Extended Observability Certification and Optimal Design of Moving Horizon Estimators. IEEE Transactions on Automatic Control, Volume 67, Issue 7, pages 3663-3669, July 2022.

Options

Extend the state $\rightarrow$ Ext-MHE $\rightarrow \hat{x}, \hat{p} \rightarrow \hat{z}$
CPU-expensive | full estimation | non observability

Stochastic MHE $\rightarrow \hat{x}$:=argmin $\mathbb{E}_{\mathcal{P}}(\cdot)$
CPU-expensive | full estimation

This contribution

Design $\mathcal{O}$ such that:

$$\boxed{\hat{z}_k \approx \mathcal{O}(\mathbf{y}_k^{(-)})}$$

$$\text{where} \quad \mathbf{y}_k^{(-)} := \begin{bmatrix} y_k \\ y_{k-m} \\ \vdots \\ y_{k-Nm} \end{bmatrix} \in \left[\mathbb{R}^{n_y}\right]^{N+1}$$

Problem statement

Class of Systems

$$\dot{x} = f(x,u,p) \quad | \quad y = h(x,u,p)$$

$\rightarrow$ f, h known.
$\rightarrow$ p, uncertain | known statistics $\mathcal{P}$.
$\rightarrow$ y measurement vector **including u**

Partial observation target

$$z = T(x,u,p)$$

$\rightarrow$ T known.
$\rightarrow$ Example: Control law: $z = k(x,p)$

MA. Partial Extended Observability Certification and Optimal Design of Moving Horizon Estimators. IEEE Transactions on Automatic Control, Volume 67, Issue 7, pages 3663-3669, July 2022.

Options

Extend the state $\rightarrow$ Ext-MHE $\rightarrow \hat{x}, \hat{p} \rightarrow \hat{z}$
CPU-expensive | full estimation | non observability

Stochastic MHE $\rightarrow \hat{x}$:=arg**min** $\mathbb{E}_{\mathcal{P}}(\cdot)$
CPU-expensive | full estimation

This contribution

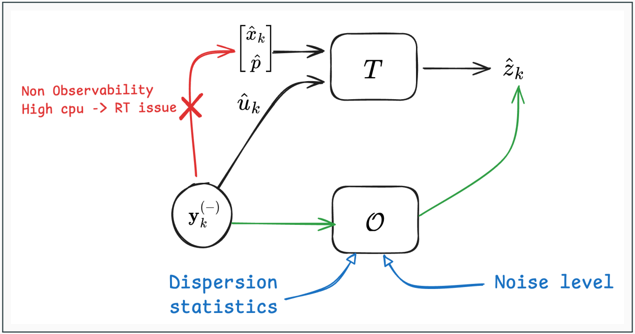
Design $\mathcal{O}$ such that:

$$\boxed{\hat{z}_k \approx \mathcal{O}(\mathbf{y}_k^{(-)})}$$

$\mathcal{O}$ must take into account:
1) The statistics $\mathcal{P}$,
2) The noise level

$$\text{where} \quad \mathbf{y}_k^{(-)} := \begin{bmatrix} y_k \\ y_{k-m} \\ \vdots \\ y_{k-Nm} \end{bmatrix} \in \left[\mathbb{R}^{n_y}\right]^{N+1}$$

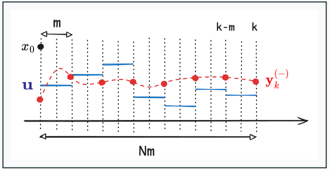
The rationale



The past measurement might retrieve the necessary information to reconstruct $z = T(x, u, p) \in \mathbb{R}$ but not necessarily $(\hat{x}, \hat{p}) \in \mathbb{R}^{n_x+n_p}$.

- ✓ Choose noise|uncertainty levels (ν, δ_p)
- ✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



- ✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \Big\{ \big(\mathbf{y}_k^{(-)}(s) + \nu\epsilon, z_k(s)\big) \Big\}_{s,k}$$

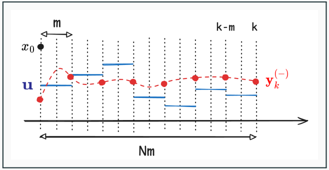
- ✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

✓ Choose noise|uncertainty levels (ν, δ_p)

✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \Big\{ \big(y_k^{(-)}(s) + \nu\epsilon, z_k(s)\big) \Big\}_{s,k}$$

✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

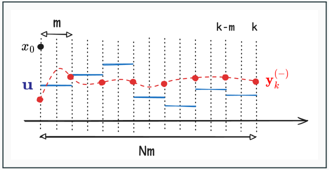
Compare the performance on *unseen* new scenarios between:

- `Random Forest`
- `XG-Boost`
- `K-Nearest-Neighbors`
- `Deep Neural Network`
- **Plars**: Sparse polynomial (this contribution).

✓ Choose noise|uncertainty levels (ν, δ_p)

✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \Big\{ \big(\mathrm{y}_k^{(-)}(s) + \nu\epsilon, z_k(s)\big) \Big\}_{s,k}$$

✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

Compare the performance on *unseen* new scenarios between:

- `Random Forest`
- `XG-Boost`
- `K-Nearest-Neighbors`
- `Deep Neural Network`
- **Plars**: Sparse polynomial (this contribution).

4.1 Electronic throttle controlled system

The first example concerns the automotive Electronic Throttle Control (**ETC**) system given by (Conatser et al., 2004):

$$\dot{x}_1 = x_2 \tag{10a}$$

$$\dot{x}_2 = \frac{1}{N_m^2 J_m + J_g}\Big[\phi(x,p) + N_m K_t x_3\Big] \tag{10b}$$

$$\dot{x}_3 = \frac{1}{L_a}\Big[-N_m K_b x_2 - R_a x_3 + u\Big] \tag{10c}$$

where the state vector is given by $x := (\theta, \dot{\theta}, e_a)$ with θ standing for the air admission angle and e_a refers to the electromotor torque induced by the current $u = i_a$ (control input). The vector $p = (N_m, J_m, J_g, \dots)$ gathers all the parameters involved in (10) (see Table 1 for the values). The nonlinear map $\phi(x,p)$ appearing in (10b) is given by:

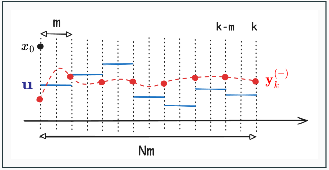
$$\phi(x,p) := -K_{sp}(x_1 - \pi/2) - (N_m^2 b_m + b_t)x_2 - \\ - 2P_{\text{atm}}(\pi - x_1)R_p^2 R_{af}\cos^2(x_1), \tag{11}$$

Building the map $\mathcal{O} : \mathrm{y}_k^{(-)} \to \hat{z}_k$ and evaluate performance

✓ Choose noise|uncertainty levels (ν, δ_p)

✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \Big\{ \big(\mathrm{y}_k^{(-)}(s) + \nu\epsilon, z_k(s)\big) \Big\}_{s,k}$$

✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

Compare the performance on *unseen* new scenarios between:

- Random Forest
- XG-Boost
- K-Nearest-Neighbors
- Deep Neural Network
- **Plars**: Sparse polynomial (this contribution).

4.2 The Lorentz oscillator

This is a widely known three dimensional nonlinear system given by:

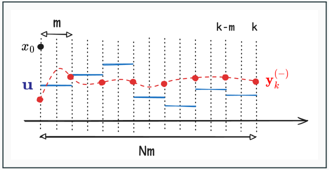
$$\dot{x}_1 = p_1(x_2 - x_1) \tag{14a}$$
$$\dot{x}_2 = x_1(p_2 - x_3) - x_2 \tag{14b}$$
$$\dot{x}_3 = x_1 x_2 - p_3 x_3 \tag{14c}$$

where the nominal parameter $p = (10, 28, 3.34)$ is used hereafter. Only the first state $y = x_1$ is supposed to be measured leaving two candidate observation targets $z_1 = x_2$ and $z_2 = x_3$. A sampling period of $\tau = 10^{-2}$ is used.

✓ Choose noise|uncertainty levels (ν, δ_p)

✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \Big\{ \big(\mathbf{y}_k^{(-)}(s) + \nu\epsilon, z_k(s)\big) \Big\}_{s,k}$$

✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

Compare the performance on *unseen* new scenarios between:

- `Random Forest`
- `XG-Boost`
- `K-Nearest-Neighbors`
- `Deep Neural Network`
- **Plars**: Sparse polynomial (this contribution).

ETC

$\mathbf{y}_k^{(-)} \in \mathbb{R}^{30} \mid y = x_1 \mid z \in \{x_2, x_3\}$

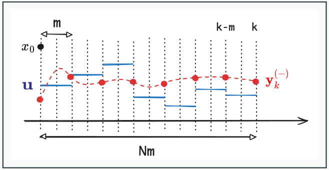
6800 samples $\leftarrow$ training

136000 samples $\leftarrow$ test

✓ Choose noise|uncertainty levels (ν, δ_p)

✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \Big\{ (\mathbf{y}_k^{(-)}(s) + \nu\epsilon, z_k(s)) \Big\}_{s,k}$$

✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

Compare the performance on *unseen* new scenarios between:

- `Random Forest`
- `XG-Boost`
- `K-Nearest-Neighbors`
- `Deep Neural Network`
- **Plars**: Sparse polynomial (this contribution).

ETC

$\mathbf{y}_k^{(-)} \in \mathbb{R}^{30} \mid y = x_1 \mid z \in \{x_2, x_3\}$

6800 samples ← training

136000 samples ← test

Lorentz

$\mathbf{y}_k^{(-)} \in \mathbb{R}^{5} \mid y = x_1 \mid z \in \{x_2, x_3\}$

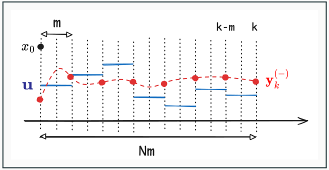
2200 samples ← training

44000 samples ← test

✓ Choose noise|uncertainty levels (ν, δ_p)

✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \Big\{ \big(y_k^{(-)}(s) + \nu\epsilon, z_k(s)\big) \Big\}_{s,k}$$

✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

Compare the performance on *unseen* new scenarios between:

- `Random Forest`
- `XG-Boost`
- `K-Nearest-Neighbors`
- `Deep Neural Network`
- **Plars**: Sparse polynomial (this contribution).

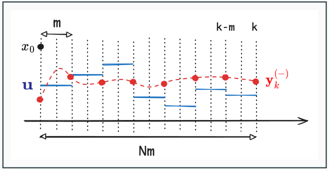
Parameter	ETC	Lorentz
n	100	250
σ_p	$\in \{0, 0.05, 0.1\}$	$\in \{0, 0.05, 0.1\}$
x_{min}	[-0.5, -0.5, -0.5]	[-1,-1,-1]
x_{max}	[0.5, 0.5, 0.5]	[1,1,1]
t_f	3.0	4.0
N	15	5
m	2	10

Table 2. Parameters used in data generation for each pair of (z, σ_p) defining the observation target and the level of parameter dispersion.

✓ Choose noise|uncertainty levels (ν, δ_p)

✓ Simulate n_{sc} scenarios (p sampled using $\mathcal{P}$)

$$s := (x_0, \mathbf{u}, p) \in \mathbb{X} \times \mathbb{U}^N \times \mathbb{P}(\delta_p)$$



✓ Collect the working data:

$$\mathcal{D}(\nu, \delta_p) := \left\{ \left(y_k^{(-)}(s) + \nu\epsilon, z_k(s)\right) \right\}_{s,k}$$

✓ Fit a model $\mathcal{O}(\nu, \delta_p)$ using $\mathcal{D}(\nu, \delta_p)$.

MA. Scalable Sparse Identification of Nonlinear Relationships. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

Compare the performance on *unseen* new scenarios between:

- `Random Forest`
- `XG-Boost`
- `K-Nearest-Neighbors`
- `Deep Neural Network`
- **Plars**: Sparse polynomial (this contribution).

Unfair comparison 🙂

For all competing solutions (except Plars), sets of hyper-parameters settings have been spanned and the ones leading to the best performance **(on test dataset)** have been used in the displayed performance results.

Presentation of the performance results

Results for ETC with $\mathbf{z} = \mathbf{x_2}$, $\sigma_{\mathbf{p}} = \mathbf{0.}$ and noise=0.0.

q	plars	RF	XGB	KNN	DNN
50%					
80%					
95%					
99%					

$$100 \times \left[\frac{p_q^{\texttt{algo}} - p_q^{\texttt{plars}}}{p^{\texttt{plars}} + \epsilon}\right] \quad ; \quad \epsilon = 0.001$$

$$p_q^{\texttt{plars}} := \frac{\texttt{percentile}(|z - \hat{z}|, q)}{\texttt{median}(|z|)}$$

Results

Results for `ETC` **with $\mathbf{z}=\mathbf{x_2}$, $\sigma_\mathbf{p}=\mathbf{0.}$ and** `noise`**=0.0.**

q	plars	RF	XGB	KNN	DNN
50%					
80%					
95%					
99%					

$$100\times\left[\frac{p_q^{\texttt{algo}}-p_q^{\texttt{plars}}}{p^{\texttt{plars}}+\epsilon}\right]\quad ;\quad \epsilon=0.001$$

$$p_q^{\texttt{plars}}:=\frac{\texttt{percentile}(|z-\hat{z}|,q)}{\texttt{median}(|z|)}$$

Results for `ETC` **with $\mathbf{z}=\mathbf{x_2}$, $\sigma_\mathbf{p}=\mathbf{0.}$ and** `noise`**=0.0.**

q	plars	RF	XGB	KNN	DNN
50%	0.0	17000.0%	10600.0%	17200.0%	93400.0%
80%	0.0	23900.0%	13450.0%	24850.0%	131950.0%
95%	0.0	20140.0%	11160.0%	22200.0%	103140.0%
99%	0.01	22278.0%	11189.0%	23300.0%	83167.0%

Results for `ETC` **with $\mathbf{z}=\mathbf{x_2}$, $\sigma_\mathbf{p}=\mathbf{0.0}$ and** `noise`**=0.025.**

q	plars	RF	XGB	KNN	DNN
50%	0.04	383.0%	214.0%	383.0%	2350.0%
80%	0.1	409.0%	197.0%	423.0%	2705.0%
95%	0.24	329.0%	146.0%	363.0%	2197.0%
99%	0.51	298.0%	109.0%	318.0%	1376.0%

Results for `ETC` **with $\mathbf{z}=\mathbf{x_2}$, $\sigma_\mathbf{p}=\mathbf{0.0}$ and** `noise`**=0.05.**

q	plars	RF	XGB	KNN	DNN
50%	0.06	203.0%	108.0%	197.0%	1541.0%
80%	0.16	220.0%	99.0%	225.0%	1648.0%
95%	0.38	179.0%	70.0%	199.0%	1381.0%
99%	0.78	169.0%	46.0%	171.0%	943.0%

Fig. 1. Partial estimation results (on unseen test data) for the observation target $z=x_2$ of `ETC` system with no parametric dispersion and different levels of noise.

Results

Results for ETC with $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = 0.$ and noise=0.0.

q	plars	RF	XGB	KNN	DNN
50%					
80%					
95%					
99%					

$$100 \times \left[\frac{p_q^{\texttt{algo}} - p_q^{\texttt{plars}}}{p^{\texttt{plars}} + \epsilon}\right] \quad ; \quad \epsilon = 0.001$$

$$p_q^{\texttt{plars}} := \frac{\texttt{percentile}(|z-\hat{z}|, q)}{\texttt{median}(|z|)}$$

Results for ETC with $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = 0.05$ and noise=0.025.

q	plars	RF	XGB	KNN	DNN
50%	0.06	219.0%	125.0%	246.0%	1539.0%
80%	0.16	202.0%	101.0%	226.0%	1525.0%
95%	0.35	174.0%	80.0%	206.0%	1299.0%
99%	0.59	182.0%	82.0%	215.0%	1081.0%

Results for ETC with $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = 0.1$ and noise=0.025.

q	plars	RF	XGB	KNN	DNN
50%	0.1	145.0%	74.0%	164.0%	829.0%
80%	0.28	127.0%	56.0%	144.0%	859.0%
95%	0.58	122.0%	56.0%	139.0%	781.0%
99%	0.93	131.0%	67.0%	143.0%	655.0%

Results for ETC with $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = 0.05$ and noise=0.05.

q	plars	RF	XGB	KNN	DNN
50%	0.08	155.0%	88.0%	173.0%	1140.0%
80%	0.2	141.0%	66.0%	160.0%	1144.0%
95%	0.44	118.0%	50.0%	143.0%	987.0%
99%	0.78	114.0%	47.0%	141.0%	775.0%

Results for ETC with $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = 0.1$ and noise=0.05.

q	plars	RF	XGB	KNN	DNN
50%	0.12	118.0%	61.0%	125.0%	775.0%
80%	0.31	105.0%	47.0%	116.0%	737.0%
95%	0.65	98.0%	47.0%	112.0%	653.0%
99%	1.06	102.0%	50.0%	117.0%	579.0%

Fig. 2. Partial estimation results (on unseen test data) for the observation target $z = x_2$ of ETC system with different levels of parametric dispersion and measurement noise.

Results

Results for `ETC` **with $\mathbf{z} = \mathbf{x_2}$, $\sigma_\mathbf{p} = \mathbf{0.}$ and** `noise`**=0.0.**

q	plars	RF	XGB	KNN	DNN
50%					
80%					
95%					
99%					

$$100 \times \left[\frac{p_q^{\texttt{algo}} - p_q^{\texttt{plars}}}{p^{\texttt{plars}} + \epsilon}\right] \quad ; \quad \epsilon = 0.001$$

$$p_q^{\texttt{plars}} := \frac{\texttt{percentile}(|z - \hat{z}|, q)}{\texttt{median}(|z|)}$$

Results for `lorentz` **with $\mathbf{z} = \mathbf{x_2}$, $\sigma_\mathbf{p} = \mathbf{0.05}$ and** `noise`**=0.025.**

q	plars	RF	XGB	KNN	DNN
50%	0.03	59.0%	28.0%	52.0%	3383.0%
80%	0.06	60.0%	34.0%	53.0%	1950.0%
95%	0.1	63.0%	71.0%	63.0%	1190.0%
99%	0.18	121.0%	145.0%	75.0%	1390.0%

Results for `lorentz` **with $\mathbf{z} = \mathbf{x_2}$, $\sigma_\mathbf{p} = \mathbf{0.1}$ and** `noise`**=0.025.**

q	plars	RF	XGB	KNN	DNN
50%	0.03	46.0%	31.0%	62.0%	3581.0%
80%	0.05	49.0%	38.0%	64.0%	2340.0%
95%	0.1	59.0%	61.0%	73.0%	1458.0%
99%	0.18	120.0%	130.0%	104.0%	1370.0%

Results for `lorentz` **with $\mathbf{z} = \mathbf{x_2}$, $\sigma_\mathbf{p} = \mathbf{0.05}$ and** `noise`**=0.05.**

q	plars	RF	XGB	KNN	DNN
50%	0.05	14.0%	14.0%	20.0%	1988.0%
80%	0.1	16.0%	14.0%	21.0%	1155.0%
95%	0.16	25.0%	20.0%	22.0%	738.0%
99%	0.27	76.0%	64.0%	24.0%	975.0%

Results for `lorentz` **with $\mathbf{z} = \mathbf{x_2}$, $\sigma_\mathbf{p} = \mathbf{0.1}$ and** `noise`**=0.05.**

q	plars	RF	XGB	KNN	DNN
50%	0.05	13.0%	13.0%	20.0%	2017.0%
80%	0.09	11.0%	13.0%	19.0%	1300.0%
95%	0.16	13.0%	20.0%	18.0%	836.0%
99%	0.28	33.0%	62.0%	41.0%	793.0%

Fig. 4. Partial estimation results (on unseen test data) for the observation target $z = x_2$ of `Lorentz` system with different levels of parametric dispersion and measurement noise.

Results

Results for `ETC` **with** $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = \mathbf{0.}$ **and** `noise`**=0.0.**

q	plars	RF	XGB	KNN	DNN
50%					
80%					
95%					
99%					

$$100 \times \left[\frac{p_q^{\texttt{algo}} - p_q^{\texttt{plars}}}{p^{\texttt{plars}} + \epsilon}\right] \quad ; \quad \epsilon = 0.001$$

$$p_q^{\texttt{plars}} := \frac{\texttt{percentile}(|z - \hat{z}|, q)}{\texttt{median}(|z|)}$$

Results for `lorentz` **with** $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = \mathbf{0.05}$ **and** `noise`**=0.025.**

q	plars	RF	XGB	KNN	DNN
50%	0.03	59.0%	28.0%	52.0%	3383.0%
80%	0.06	60.0%	34.0%	53.0%	1950.0%
95%	0.1	63.0%	71.0%	63.0%	1190.0%
99%	0.18	121.0%	145.0%	75.0%	1390.0%

Results for `lorentz` **with** $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = \mathbf{0.1}$ **and** `noise`**=0.025.**

q	plars	RF	XGB	KNN	DNN
50%	0.03	46.0%	31.0%	62.0%	3581.0%
80%	0.05	49.0%	38.0%	64.0%	2340.0%
95%	0.1	59.0%	61.0%	73.0%	1458.0%
99%	0.18	120.0%	130.0%	104.0%	1370.0%

Results for `lorentz` **with** $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = \mathbf{0.05}$ **and** `noise`**=0.05.**

q	plars	RF	XGB	KNN	DNN
50%	0.05	14.0%	14.0%	20.0%	1988.0%
80%	0.1	16.0%	14.0%	21.0%	1155.0%
95%	0.16	25.0%	20.0%	22.0%	738.0%
99%	0.27	76.0%	64.0%	24.0%	975.0%

Results for `lorentz` **with** $\mathbf{z = x_2}$, $\sigma_\mathbf{p} = \mathbf{0.1}$ **and** `noise`**=0.05.**

q	plars	RF	XGB	KNN	DNN
50%	0.05	13.0%	13.0%	20.0%	2017.0%
80%	0.09	11.0%	13.0%	19.0%	1300.0%
95%	0.16	13.0%	20.0%	18.0%	836.0%
99%	0.28	33.0%	62.0%	41.0%	793.0%

Fig. 4. Partial estimation results (on unseen test data) for the observation target $z = x_2$ of `Lorentz` system with different levels of parametric dispersion and measurement noise.

Dataset available @:

MA. Dataset for nonlinear state estimator design. 10.34740/KAGGLE/DSV/10016280

Full details available @:

MA. Partial state estimator design for Nonlinear Uncertain System. In Nonlinear control of Uncertain Systems. **Springer Nature** 2026.

Learning might be a way to explicitly build partial stochastic estimator in the presence of uncertainties with known dispersion.

Sparse models derivation is crucial for learning with limited data.

Since, successful extension of the the current investigation to piece-wise polynomial relationships.

Take-Home Messages

Learning might be a way to explicitly build **partial stochastic estimator** in the presence of uncertainties with known dispersion.

Sparse models derivation is crucial for learning with **limited data**.

Since, successful extension of the the current investigation to **piece-wise polynomial** relationships.

Questions?