Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Robust Sparse Identification of Nonlinear Dynamical Systems Using Huber Loss Function

Takuma Kobayashi[1†] and Ryo Tanaka[2]

[1]Advanced Engineering School, National Institute of Technology, Kurume College, Fukuoka, Japan
(Tel: +81-80-2777-9524; E-mail: a4111tk@kurume.kosen-ac.jp)
[2]Department of Control and Information Systems Engineering, National Institute of Technology,
Kurume College, Fukuoka, Japan
(Tel: +81-942-35-9372; E-mail: rtanaka@kurume-nct.ac.jp)

Abstract: This paper proposes a robust extension of the Sparse Identification of Nonlinear Dynamical Systems (SINDy) framework by incorporating the Huber loss function. The standard SINDy approach, such as STLS and Lasso, employ the L2 norm loss for the error term and are known to be vulnerable to outliers. To address this problem, the Huber loss, which combines the benefits of L2 and L1 norms, has been adopted to ensure robustness against both small errors and large outliers. Since real-world measurement noise is typically Gaussian, the Huber loss is considered well suited for SINDy. To evaluate the proposed method, numerical experiments were carried out on two systems: the Lorenz system without control input, and a two-dimensional drone model with control input. In both cases, noise was synthetically added into training data so that 80% of the data is corrupted with relatively small noise and 20% with larger noise, simulating practical measurement conditions with occasional outliers. The results show that the proposed method achieves higher accuracy in model identification compared to the conventional SINDy under such mixed-noise conditions.

Keywords: Data-driven modeling, Sparse identification, Nonlinear systems, Huber loss function, Robust regression

1. INTRODUCTION

Data-driven modeling has attracted considerable attention as an approach that enables system control and analysis without explicit knowledge of the governing equations [1]. Various data-driven methods, such as neural networks (NN) and dynamic mode decomposition (DMD) [2], have been proposed and reported to be effective for analyzing complex dynamical and nonlinear systems [3]. However, the models obtained from these methods are black-box models with limited interpretability. Sparse Identification of Nonlinear Dynamical systems (SINDy) [4] has recently attracted attention as a machine learning model that solves this interpretability problem. SINDy applies sparse modeling to system identifications to extract interpretable models from the data. Measured data may contain outliers due to disturbances. However, Sparse Transformed Least Squares (STLS) [4] and Lasso [5], which are typical optimization methods for SINDy, employ the L2 norm as the error term, making them susceptible to outliers. To overcome this weakness, this study examines the construction of SINDy using Huber loss [6] as the error term, and aims to propose a SINDy that is more robust to outliers.

2. METHODOLOGY

2.1. Review of SINDy

The SINDy framework is based on the following nonlinear dynamical system [4]:

$$\frac{d}{dt}\boldsymbol{x}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{1}$$

where $\boldsymbol{x}(t) = [x_1(t)\ x_2(t)\ \cdots\ x_n(t)]^\top \in \mathbb{R}^n$ is the state vector and $\boldsymbol{u}(t) = [u_1(t)\ u_2(t) \cdots\ u_r(t)]^\top \in \mathbb{R}^r$ is the control input.

The following data matrices are formed by collecting m snapshots from Eq. (1).

$$\boldsymbol{X} = [\boldsymbol{x}(t_1)\ \boldsymbol{x}(t_2)\ \cdots\ \boldsymbol{x}(t_m)]^\top, \tag{2}$$

$$\dot{\boldsymbol{X}} = [\dot{\boldsymbol{x}}(t_1)\ \dot{\boldsymbol{x}}(t_2)\ \cdots\ \dot{\boldsymbol{x}}(t_m)]^\top, \tag{3}$$

$$\boldsymbol{U} = [\boldsymbol{u}(t_1)\ \boldsymbol{u}(t_2)\ \cdots\ \boldsymbol{u}(t_m)]^\top. \tag{4}$$

Here, $\dot{\boldsymbol{X}}$ denotes the time derivative of the state data $\boldsymbol{X}$.

Next, a dictionary matrix $\boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})$ is constructed from both linear and nonlinear functions of $\boldsymbol{X}$ and $\boldsymbol{U}$. Using the dictionary matrix $\boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})$, Eq. (1) is approximated as follows:

$$\dot{\boldsymbol{X}} \approx \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\Xi, \tag{5}$$

where $\Xi \in \mathbb{R}^{p\times n}$ is the coefficient matrix, $\Xi = [\boldsymbol{\xi}_1\ \boldsymbol{\xi}_2\ \cdots\ \boldsymbol{\xi}_n]$, and $\boldsymbol{\xi}_k (k = 1, \cdots, n) \in \mathbb{R}^p$ is the coefficient of the dictionary function. By improving the sparsity of Ξ, an approximate model fitting Eq. (1) can be obtained. In practice, each column $\boldsymbol{\xi}_k$ of Ξ is obtained by solving the following sparse regression problem:

$$\boldsymbol{\xi}_k = \arg\min_{\boldsymbol{\xi}_k} \left\| \dot{\boldsymbol{X}}_k - \Theta(\boldsymbol{X}, \boldsymbol{U})\, \boldsymbol{\xi}_k \right\|_2^2 + \lambda \|\boldsymbol{\xi}_k\|_1, \tag{6}$$

where $\dot{\boldsymbol{X}}_k$ denotes the k-th column of $\dot{\boldsymbol{X}}$, $\|\cdot\|_2$ denotes a L2 norm, $\|\cdot\|_1$ denotes a L1 norm, and $\lambda > 0$ is a regularization parameter promoting sparsity.

---

† Takuma Kobayashi is the presenter of this paper.

978-4-9077-6487-6 PR0001/25 ¥400 © 2025 SICE

2.2. Huber Loss Function

The Huber loss function is defined as follows [6]:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & |a| \le \delta \\ \delta(|a| - \frac{1}{2}\delta) & |a| > \delta. \end{cases} \tag{7}$$

If the absolute value of the error is less than a certain threshold δ, it acts as the L2 norm, whereas if it exceeds δ, it acts as the L1 norm. This property makes it robust to outliers.

2.3. Proposed Method

In the conventional formulation Eq. (6), the residual term is using the L2 norm, which makes the method sensitive to outliers in the data. To address this issue, this study proposes replacing L2 loss with the Huber loss in the error term, thereby constructing a more robust variant of SINDy against outliers. By applying the Huber loss L_δ, the optimization problem to be solved becomes

$$\boldsymbol{\xi}_k = \arg\min_{\boldsymbol{\xi}_k} L_\delta\Big(\dot{\boldsymbol{X}}_k - \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\boldsymbol{\xi}_k\Big) + \lambda \left\|\boldsymbol{\xi}_k\right\|_1 . \tag{8}$$

2.4. Solution Algorithm

In this work, Alternating Direction Method of Multipliers (ADMM) [7] is used to solve Eq. (8). To apply ADMM, Eq. (8) is rewritten in the following constrained form, following the formulation proposed in [8].

$$\begin{aligned} &\arg\min_{\boldsymbol{r}_k, \boldsymbol{z}_k} L_\delta\big(\boldsymbol{r}_k\big) + \lambda \left\|\boldsymbol{z}_k\right\|_1 , \\ \text{s.t.} \ &\begin{pmatrix} \boldsymbol{r}_k \\ \boldsymbol{z}_k \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{X}}_k - \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\boldsymbol{\xi}_k \\ \boldsymbol{\xi}_k \end{pmatrix}. \end{aligned} \tag{9}$$

Applying ADMM to Eq. (9) yields the following iterative updates, based on the approach introduced in [8].

$$\begin{aligned} \boldsymbol{\xi}_k^{(i+1)} &= \arg\min_{\boldsymbol{\xi}_k} \left\| \boldsymbol{r}_k^{(i)} - [\dot{\boldsymbol{X}}_k - \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\boldsymbol{\xi}_k^{(i)}] + \boldsymbol{U}_{r,k}^{(i)} \right\|_2^2 \\ &\quad + \left\| \boldsymbol{z}_k^{(i)} - \boldsymbol{\xi}_k^{(i)} + \boldsymbol{U}_{z,k}^{(i)} \right\|_2^2 \\ &= \Big(\boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})^\top \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U}) + \boldsymbol{I}\Big)^{-1} \\ &\quad \Big\{ \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})^\top \big(\dot{\boldsymbol{X}}_k - \boldsymbol{r}_k^{(i)} - \boldsymbol{U}_{r,k}^{(i)}\big) + \boldsymbol{z}_k^{(i)} + \boldsymbol{U}_{z,k}^{(i)} \Big\}, \end{aligned} \tag{10}$$

$$\begin{aligned} \boldsymbol{r}_k^{(i+1)} &= \arg\min_{\boldsymbol{r}_k} L_\delta\big(\boldsymbol{r}_k^{(i)}\big) \\ &\quad + \frac{\rho}{2} \left\| \boldsymbol{r}_k^{(i)} - [\dot{\boldsymbol{X}}_k - \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\boldsymbol{\xi}_k^{(i+1)}] + \boldsymbol{U}_{r,k}^{(i)} \right\|_2^2 \\ &= \operatorname{prox}_{\frac{1}{\rho}L_\delta(\cdot)} \big(\dot{\boldsymbol{X}}_k - \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\boldsymbol{\xi}_k^{(i+1)} - \boldsymbol{U}_{r,k}^{(i)}\big) \\ &= \operatorname{prox}_{\frac{1}{\rho}L_\delta(\cdot)} \big(\boldsymbol{a}_k^{(i)}\big) \\ &= \begin{cases} \operatorname{sign}\big(\boldsymbol{a}_k^{(i)}\big) \odot \max\big(0, |\boldsymbol{a}_k^{(i)}| - \frac{1}{\rho}\big) & |\boldsymbol{a}_k^{(i)}| > \delta \\ \frac{\rho}{1+\rho}\big(\boldsymbol{a}_k^{(i)}\big) & |\boldsymbol{a}_k^{(i)}| \le \delta, \end{cases} \end{aligned} \tag{11}$$

where $\boldsymbol{a}_k^{(i)} \triangleq \dot{\boldsymbol{X}}_k - \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\boldsymbol{\xi}_k^{(i+1)} - \boldsymbol{U}_{r,k}^{(i)}$ is the residual inside the proximal operator.

$$\begin{aligned} \boldsymbol{z}_k^{(i+1)} &= \arg\min_{\boldsymbol{z}_k} \lambda \left\| \boldsymbol{z}_k^{(i)} \right\|_1 \\ &\quad + \frac{\rho}{2} \left\| \boldsymbol{z}_k^{(i)} - \big(\boldsymbol{\xi}_k^{(i+1)} - \boldsymbol{U}_{z,k}^{(i)}\big) \right\|_2^2 \\ &= \operatorname{prox}_{\frac{\lambda}{\rho}\|\cdot\|_1} \big(\boldsymbol{\xi}_k^{(i+1)} - \boldsymbol{U}_{z,k}^{(i)}\big) \\ &= \operatorname{sign}\big(\boldsymbol{\xi}_k^{(i+1)} - \boldsymbol{U}_{z,k}^{(i)}\big) \\ &\quad \odot \max\Big(0, \left\| \xi_k^{(i+1)} - \boldsymbol{U}_{z,k}^{(i)} \right\|_1 - \frac{\lambda}{\rho}\Big), \end{aligned} \tag{12}$$

$$\boldsymbol{U}_{r,k}^{(i+1)} = \boldsymbol{U}_{r,k}^{(i)} + \boldsymbol{r}_k^{(i+1)} - \big(\dot{\boldsymbol{X}}_k - \boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})\boldsymbol{\xi}_k^{(i+1)}\big), \tag{13}$$

$$\boldsymbol{U}_{z,k}^{(i+1)} = \boldsymbol{U}_{z,k}^{(i)} + \boldsymbol{z}_k^{(i+1)} - \boldsymbol{\xi}_k^{(i+1)}. \tag{14}$$

Here, $\odot$ denotes the Hadamard product, and ”prox” denotes the proximal operator [9].

3. NUMERICAL SIMULATIONS

In this section, the effectiveness of the proposed Huber-SINDy method is evaluated by applying it to two representative nonlinear systems. In this study, true derivatives are first computed from the system's dynamics, and noise is the added. This approach eliminates errors introduced by numerical differentiation and allows us to isolate the effect of measurement noise on coefficient estimation. An alternative way to avoid numerical differentiation errors is to formulate the problem in a discrete-time framework. However, this approach makes visual comparison of the experimental results more difficult, so it is not adopted here. Moreover, methods such as total-varitaion-regularized derivatives [10] could be considered to further mitigate the impact of numerical differentiatoin errors.

Following the methodology of [11], Gaussian noise with variance σ_1^2 is added to 80% of the data to simulate typical measurement errors, and Gaussian noise with a larger variance σ_2^2 is added to the remaining 20% to reproduce outliers.

Furthermore, in both cases, numerical integration was performed using the classical fourth-order Runge–Kutta method.

3.1. Lorenz System

First, as a test case for uncontrolled dynamics, the Lorenz system [12] without any control inputs is considered. The Lorenz system is a classic example of chaotic dynamics governed by

$$\begin{cases} \dfrac{dx}{dt} = -\sigma(y - x), & (15) \\ \dfrac{dy}{dt} = x(\rho - z) - y, & (16) \\ \dfrac{dz}{dt} = xy - \beta z. & (17) \end{cases}$$

The parameters of the Lorenz system are set to $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. The initial state is $\boldsymbol{x}(0) = [0\ 1\ 20]^\top$. A sampling interval of $\Delta t = 0.01$[s] is used. The library matrix $\Theta(\boldsymbol{X})$ is constructed using only polynomial basis functions up to degree $d = 2$.

3.2. Two-dimensional drone

Next, as a test case with control inputs, a two-dimensional drone model [13] is considered. A two-dimensional drone model is used as an example with control input because it is low-dimensional yet exhibits complex nonlinearities.

$$\begin{cases} m\ddot{x} = -(f_1 + f_2)\sin\theta, & (18) \\ m\ddot{y} = (f_1 + f_2)\cos\theta - mg, & (19) \\ J\ddot{\theta} = l(f_1 - f_2). & (20) \end{cases}$$

Here, (x, y) denote the horizontal and vertical position coordinates of the drone, θ is pitch angle, m is mass, l is the distance from the center of mass to the rotors, J is moment of inertia about the pitch axis, and g is the gravitational acceleration. In this simulation, these parameters are set to $m = 1.0$[kg], $l = 0.086$[m], $J = 0.1[\mathrm{kg} \cdot \mathrm{m}^2]$, and $g = 9.81[\mathrm{m/s}^2]$. A sampling interval of $\Delta t = 0.01$[s] is used. The dictionary matrix $\boldsymbol{\Theta}(\boldsymbol{X}, \boldsymbol{U})$ is constructed from the following candidate basis functions:

$$\{\ 1,\ \dot{x},\ \dot{y},\ \dot{\theta},\ (f_1 + f_2)\sin\theta,\ (f_1 + f_2)\cos\theta,\ f_1 - f_2\}.$$

3.3. Parameter Setting

The setting parameters used in this study are listed in Table 1. The penalty parameters of ADMM were set to 1.0 to ensure the numerical stability. A convergence tolerance and a maximum number of iterations were chosen to ensure both accuracy and reasonable computational cost. The threshold of STLS was set to 0.05 to optimize model sparsity and identification accuracy. The noise parameters σ_1 and σ_2 were set based on the related work [11].

Table 1 Parameters of algorithm and noise

Category	Parameter	Value
ADMM	Penalty parameters ρ, λ	1.0, 1.0
	Max iterations K_{max}	5000
	Convergence tol. ε	10^{-6}
STLS	Threshold λ	0.05
	Max iterations K_{max}	5000
Noise	Small-noise std. σ_1	10^{-3}
	Large-noise std. σ_2	0.5

4. RESULT AND DISCUSSION

4.1. Objective evaluation

To evaluate the performance of the estimated coefficient matrix, this study employs the Integral of Absolute Error (IAE) metric. Let the true coefficient matrix be Ξ and the estimated coefficient matrix be $\hat{\Xi}$. Then, the IAE is calculated as

$$\mathrm{IAE} = \sum_{i=1}^{p} \sum_{j=1}^{n} \left| \Xi_{ij} - \hat{\Xi}_{ij} \right|. \tag{21}$$

By definition, a lower IAE value corresponds to better model performance.

4.2. Results

In this section, the performance of the proposed method is evaluated against the conventional method based on the IAE metric.

First, the performance comparison results for the Lorenz system are presented in Tables 2 to 5.

Table 2 Coefficient comparison for Eq. (15)

Basis	True	Conventional	Proposed
1	0	2.14144e-03	2.61026e-03
x	-10	-9.07281	-9.99739
y	10	9.44365	9.99837
z	0	-4.31789e-02	-4.56194e-04
xx	0	-1.90639e-02	0.00000
xy	0	1.44366e-02	0.00000
xz	0	-2.37843e-02	0.00000
yy	0	-1.37063e-03	0.00000
yz	0	1.31139e-02	0.00000
zz	0	2.30649e-03	0.00000
IAE	-	1.17239	1.64900e-05

Table 3 Coefficient comparison for Eq. (16)

Basis	True	Conventional	Proposed
1	0	-0.189081	-3.95669e-03
x	28	27.2816	27.9987
y	-1	-0.568924	-0.999078
z	0	3.50579e-02	4.17531e-04
xx	0	7.63396e-03	0.00000
xy	0	-8.32381e-03	0.00000
xz	-1	-0.980496	-0.999961
yy	0	2.19053e-03	0.00000
yz	0	-1.09502e-02	0.00000
zz	0	-1.29085e-03	0.00000
IAE	-	0.739606	0.999961e-05

Table 4 Coefficient comparison for Eq. (17)

Basis	True	Conventional	Proposed
1	0	2.16721	1.16896e-03
x	0	9.08482e-02	4.68215e-04
y	0	-3.16791e-02	-2.30046e-04
z	-2.66667	-2.97618	-2.66693
xx	0	-4.63305e-02	0.00000
xy	1	0.950357	1.00002
xz	0	-2.52414e-03	0.00000
yy	0	4.83118e-04	0.00000
yz	0	1.98461e-04	0.00000
zz	0	9.97369e-03	0.00000
IAE	-	4.80485	1.70912e-06

Table 5 Comparison of conventional and proposed methods

	Conventional	Proposed
IAE	6.71685	3.65371e-05

Next, the performance comparison results for the two-dimensional drone model are shown in Tables 6 to 9.

Table 6 Coefficient comparison for Eq. (18)

Basis	True	Conventional	Proposed
1	0	-0.103494	0.00000
$\dot{x}$	0	0.00000	0.00000
$\dot{y}$	0	-0.172011	0.00000
$\dot{\theta}$	0	-0.648500	0.00000
$(f_1 + f_2)\sin\theta$	-1	-0.938295	-0.99989
$(f_1 + f_2)\cos\theta$	0	0.00000	0.00000
$f_1 - f_2$	0	0.162533	0.00000
IAE	-	1.14822	1.10000e-04

Table 7 Coefficient comparison for Eq. (19)

Basis	True	Conventional	Proposed
1	-9.81	-9.16991	-9.80933
$\dot{x}$	0	5.57585e-02	0.00000
$\dot{y}$	0	8.20100e-02	0.00000
$\dot{\theta}$	0	-2.97618e-02	0.00000
$(f_1 + f_2)\sin\theta$	0	0.00000	0.00000
$(f_1 + f_2)\cos\theta$	1	0.971251	0.99994
$f_1 - f_2$	0	0.00000	0.00000
IAE	-	0.836369	7.30000e-04

Table 8 Coefficient comparison for Eq. (20)

Basis	True	Conventional	Proposed
1	0	0.00000	0.00000
$\dot{x}$	0	0.00000	0.00000
$\dot{y}$	0	0.206162	0.00000
$\dot{\theta}$	0	0.783874	1.44455e-02
$(f_1 + f_2)\sin\theta$	0	0.00000	0.00000
$(f_1 + f_2)\cos\theta$	0	0.00000	0.00000
$f_1 - f_2$	0.86	0.722301	0.856495
IAE	-	1.12773	1.79505e-02

Table 9 Comparison of conventional and proposed methods

	Conventional	Proposed
IAE	3.11231	1.87905e-02

From the experimental results, the proposed method achieves lower IAE values than the conventional approach, demonstrating superior identification accuracy.

4.3. Discussion

In the conventional method, errors are evaluated using the L2 norm, which amplifies the impact of outliers in the data. In contrast, the proposed method employs the Huber loss, applying the L1 norm to large residuals corresponding to outliers, thereby mitigating their influence compared to the L2 norm. Consequently, the proposed method achieved higher model accuracy than the conventional one.

5. CONCLUSION

In this study, the development of a more outlier-robust SINDy was investigated by employing the Huber loss in the error term. As a result, the proposed Huber-SINDy exhibited superior robustness to outliers compared to the conventional approach. Future work will focus on applying this method to real-world systems.

REFERENCES

[1] S. L. Brunton and J. N. Kutz, "Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control", Cambridge University Press, Cambridge, UK, 2019.

[2] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic Mode Decomposition with Control", *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[3] U. Fasel, E. Kaiser, J. N. Kutz, B. W. Brunton, and S. L. Brunton, "SINDy with Control: A Tutorial", arXiv:2108.13404, 2021.

[4] S.L. Brunton, J.L. Proctor, and J.N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems", Proc. Natl. Acad. Sci. U.S.A. vol. 113 no. 15, pp. 3932–3937, 2016.

[5] N. Masafumi, "Sparse Modeling: From Fundamentals to Applications in Dynamical Systems", Corona Publishing Co., Ltd., 1st ed., 5th printing, November 5, 2020.

[6] Peter J. Huber, "Robust Estimation of a Location Parameter", *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers", Foundations and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2010.

[8] B. Wohlberg, "Convolutional Sparse Representations as an Image Model for Impulse Noise Restoration", in *Proc. 2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pp. 1–5, 2016.

[9] N. Parikh and S. Boyd, "Proximal Algorithms", *Foundations and Trends in Optimization*, vol. 1, no. 3, 2013.

[10] R. Chartrand. Numerical differentiation of noisy, nonsmooth data. ISRN Applied Mathematics, 2011.

[11] X. He, Z. Sun, "Sparse identification of dynamical systems by reweighted l1-regularized least absolute deviation regression", Communications in Nonlinear Science and Numerical Simulation, vol. 131, pp. 1-23 2024.

[12] E. Lorenz, "Deterministic nonperiodic flow", *Univ Chaos*, pp. 367–378, 2017.

[13] T. Fukatsu, T. Hishinuma, and D. Aramaki, *Model Predictive Control with Python and CasADi* Kodansha, Tokyo, 1st ed., June 18, 2024 in Japanese.