# Optimising SINDy via nullcline reconstruction

Zhaoshuang.Cui, Xiaodan.Yang

July 2025

## 1 Introduction

The SINDy algorithm[2] is a nonlinear dynamic system identification approach which is based on the hypothesis that the governing equations of a nonlinear system can be given by selecting a few suitable basis functions. Consider the problem of discovering nonlinear systems of the form:

$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$,

where $\mathbf{x}(t) = [\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_n(t)]^\top \in \mathbb{R}^n$ and $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$. To identify $\mathbf{f}$ from data, we first collect time histories of the state $x(t)$. We then either directly measure its derivative $\dot{\mathbf{x}}(t)$ or approximate it numerically from the state measurements. The data, sampled at discrete times $t_1, t_2, \ldots, t_N$ , are organized into two matrices:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1)^\top \\ \mathbf{x}(t_2)^\top \\ \vdots \\ \mathbf{x}(t_\mathcal{N})^\top \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1(t_1) & \mathbf{x}_2(t_1) & \cdots & \mathbf{x}_n(t_1) \\ \mathbf{x}_1(t_2) & \mathbf{x}_2(t_2) & \cdots & \mathbf{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1(t_\mathcal{N}) & \mathbf{x}_2(t_\mathcal{N}) & \cdots & \mathbf{x}_n(t_\mathcal{N}) \end{bmatrix}$$

The derivative matrix $\mathbf{X}$ is defined analogously, with each element being the time derivative of the corresponding entry in $\mathbf{X}$. The next key building block in the SINDy algorithm is the construction of a dictionary $\Theta(\mathbf{X})$, containing candidate basis functions (e.g., constant, polynomial or trigonometric functions).

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & | & | \\ 1 & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \cdots \\ | & | & | & | & | & | & | & | \end{bmatrix}$$

We seek to identify a sparse vector $\Xi = [\xi_1, \xi_2, \ldots, \xi_n]$, where $\xi_i \in \mathbb{R}^m$ with $m$ denoting the number of columns in $\Theta$, that determines which features from the dictionary are active and their corresponding coefficients.

$\mathbf{X} = \Theta(\mathbf{X})\Xi$

To solve for the sparse coefficient matrix, we use sparse regression (e.g., Least Absolute Shrinkage and Selection Operator (LASSO):

$\min_\Xi \|X - \Theta(X)\Xi\|_2^2 + \lambda\|\Xi\|_1$

where $\lambda$ is a sparsity-promoting regularisation parameter.)

Due to the sensitivity to the initial data of dynamical systems, the error between the identified dynamic system and the original dynamic system will increase as time changes. In this report, we will introduce an optimised SINDy using nullcline reconstruction based on [4] to capture dynamic behaviour more accurately and also provide an idea about how to apply this optimised SINDy to a nullcline-based control strategy[1].

## 2 Prerequisite knowledge

1. For an n-dimensional dynamic system which is generated by a system of differential equations, the nullcline of $x_i$ is the set of points which satisfy $\frac{dx_i}{dt} = f_i(X) = 0$, where $X = (x_1, ..., x_n)$.

2. An attractor is a set of states (points in the phase space), invariant under the dynamics, towards which neighbouring states in a given basin of attraction asymptotically approach in the course of dynamic evolution.

3. Three criteria:

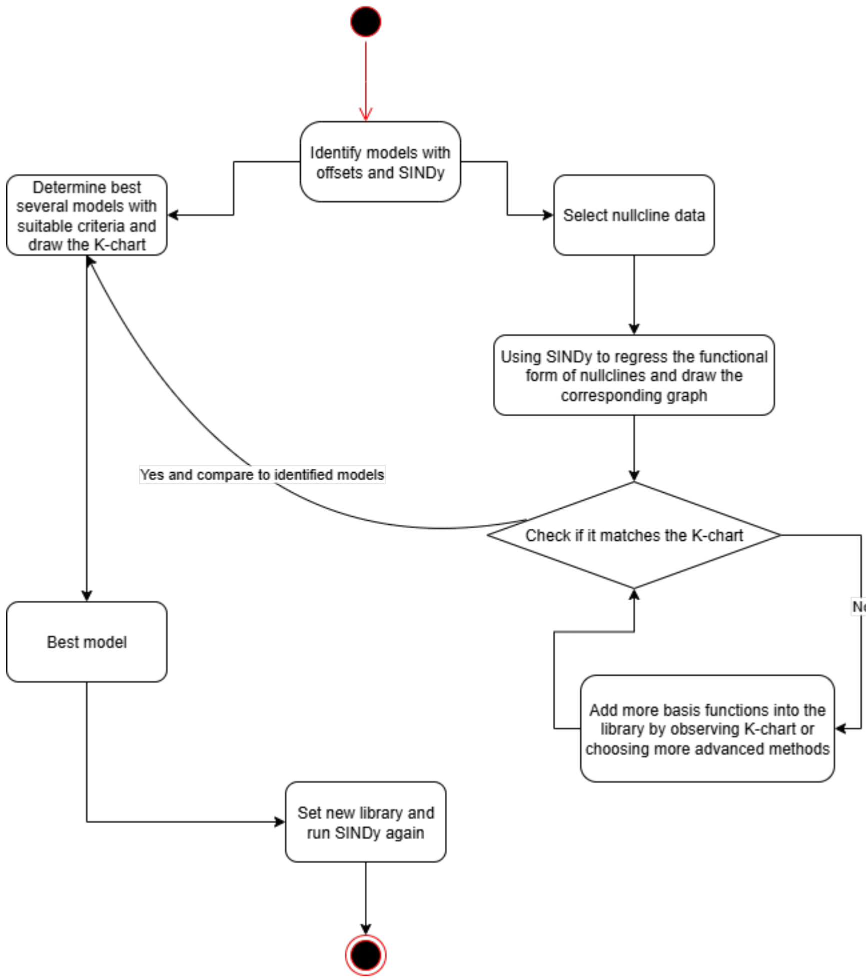
3.1. $K$ is the number of nonzero terms in the dynamic system identified according to the library, or equivalently, is the number of non-zero entries in the matrix $\boldsymbol{\xi}$(coefficient matrix in SINDy algorithm).

3.2. $R^2 = 1 - \frac{\sum_i \left(X_i - N\left[u,v,uv,u^2,v^2,...,\xi\right]\right)^2}{\sum_i \left(X_i - \bar{X}\right)^2}$, where $X$ represents measured data $\bar{X}$ represents its mean.

3.3. The distance to the attractor $\delta$ assesses whether a model can reproduce the original limit cycle after perturbation. To compute $\delta$, we simulate the model under 50 varied initial conditions $\mathbf{Y}_{\text{per}} = (w, z)$ and measure the average, relative distance to the original limit cycle data $\mathbf{X} = (u, v)$: $\delta = \frac{\sum_i \min \sqrt{(u_i - w_j)^2 + (v_i - z_j)^2}}{\sum_i (u_i - \bar{u})^2 + (v_i - \bar{v})^2}$

Based on [4], we use $(\Delta x, \Delta y)$ to represent the offset dataset $\mathcal{D} = \{(\Delta x_k, \Delta y_k) \mid k = 1, \ldots, M\}$ and apply SINDy to each offset $(\Delta x_k, \Delta y_k)$ to calculate $K$, then obtain the $K-$chart $K(\Delta x, \Delta y)$. If we add one of the offsets $(\Delta x_k, \Delta y_k)$ to our original data set $X$ and $\exists i\ st. f_i((\Delta x_k, \Delta y_k)) = 0$, the value of $K$ will drop dramatically after SINDy identifies it. Therefore, we can use the distribution of $K$ to reveal the structures of their nullclines (in the paper, the offset ranges from -2 to 2 with an equal distance of 0.1).

# 3 Algorithm




**Remark:**

Suitable criteria means that if we know $X$ comes from an attractor, then we rank models by $\delta, R^2$, and $K$, otherwise ranking by $R^2$ and $K$.

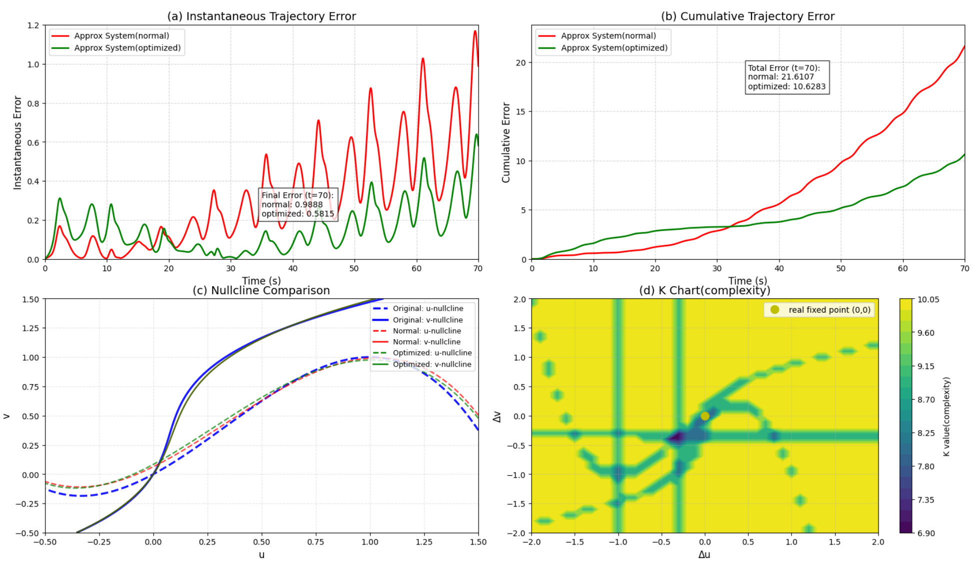
The threshold $\lambda$ in SINDy should be set relatively low since some unpredictable changes may happen in the expression of the identified systems after applying offsets.

This algorithm is only designed for two-dimensional dynamic systems.

# 4 Example

Here we present an example for this algorithm.

Let the original system be bicubic model: $\begin{cases} \frac{\mathrm{d}u}{\mathrm{d}t} = -u^3 + u^2 + u - v \\ \frac{\mathrm{d}v}{\mathrm{d}t} = \frac{-v^3}{2} + \frac{v^2}{2} - \frac{v}{3} + u \end{cases}$ and $\lambda = 0.08$. Here we only use two criteria in our algorithm, that is $R^2$ and $K$. Offset $(\Delta x, \Delta y)$ is the dataset ranges from $-2$ to 2 with equal distance 0.1. The initial data are four trajectories which start from $(0.1, 0.1)$, $(-4, 0.5)$, $(1, -3)$ and $(3, -3.5)$ respectively and end up with $t = 40$.




This graph shows that the optimised model exhibits a better approximation to the original system in the long term.

In practice, we need to choose the threshold wisely to balance whether the dynamic system contains too many minor terms and whether trajectories are captured precisely. These minor initial errors might cause significant errors with time changes, but if we use our modified SINDy, the error in the long term can be dramatically decreased, as it provides a better approximation to their nullcline structures.

# 5 Application

## 5.1 Controller(nullcline)

[1] introduced a control strategy by manipulating nullclines. The following equations can describe the FHN model: $\begin{cases} \dot{x}_1 = \frac{1}{\varepsilon}\left[\kappa\left(x_1\right) - x_2 + I\right] \\ \dot{x}_2 = x_1 + a - bx_2 \end{cases}$

where $\kappa(x_1)$ is a polynomial of third degree, $a, b, \varepsilon \in \mathbb{R}$ are constant parameters and $I \in \mathbb{R}$ is an input currentlike quantity that shifts the first nullcline (without loss of generality, we can set $I = 0$ ). By observing this ode system, we can know that it has an S-shaped nullcline and a straight nullcline. Therefore, the dynamic system generated by it will have a limit cycle, and nearby trajectories will also show similar oscillations. By controlling the parameters of its nullclines, we can control the oscillation to some extent. Because of this kind of oscillatory property, this control strategy can be implemented in robot walking or other situations.

### 5.2 PWL technique

Since $\kappa(x_1)$ is a polynomial with degree 3, it is hard to manipulate it directly. [1] introduced the PWL technique to transform it into a piecewise linear function.

$$\Pi(x) = a_0 + a_1 x + \sum_{j=1} b_j \left| x - E_j \right|$$

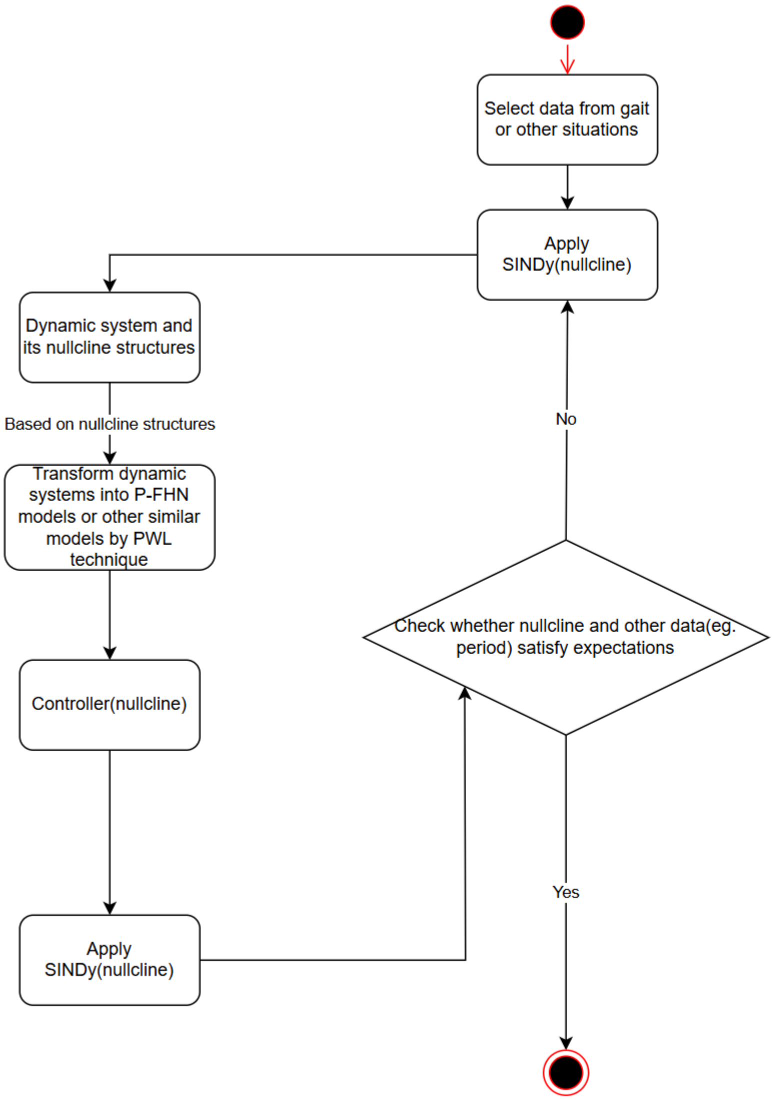
where $\Pi(0) \in \mathbb{R}$ and $a_0, a_1, b_j, E_j \in \mathbb{R}$, with $j \in \{1, \ldots, n\}$, depend on the slopes of the segments:

$$\begin{cases} a_0 = \Pi(0) - \sum_{j=1}^{n} b_j \left| E_j \right| \\ a_1 = 0.5\left(m_0 + m_n\right) \\ b_j = 0.5\left(m_j - m_{j-1}\right) \end{cases}$$

Moreover, we can set $E_j$ equal to the $j$th point such that $\frac{d\kappa}{dx_1}(E_j) = 0$, $m_j = \frac{\kappa(E_{j+1}) - \kappa(E_j)}{E_{j+1} - E_j}$ and $m_0$ and $m_n$ need to be set additionally. Since the controller in [1] is based on the FHN model, the oscillatory period mainly depends on the leftmost and rightmost parts of the S-shaped nullcline, i.e. we just need to control $m_0$ and $m_n$ properly if we want to control the oscillatory period.

### 5.3 Combine it with SINDy(nullcline)

One of the main demerits of this control strategy is the lack of flexibility. It needs the model to be the prefixed FHN models and cannot handle accidents(e.g., the robot lost one leg suddenly). We will invoke SINDy(nullcline) to modify the control process by the following procedures.




# 6 Conclusion

This report presents an optimised SINDy algorithm integrated with nullcline reconstruction, aiming to address the limitation of traditional SINDy, where the error between the identified and original dynamical systems grows over time due to sensitivity to initial data. The core value of this optimised approach lies in two aspects: first, it leverages offset datasets to generate $K$-charts, which effectively reveal nullcline structures and thereby enhance the long-term accuracy of dynamic behavior capture for two-dimensional systems; second, it provides a flexible framework for nullcline-based control strategies, overcoming the inflexibility of prefixed models (e.g., the FHN model) and enabling adaptation to unexpected scenarios such as sudden leg loss in robots. In the bicubic model example, the optimised algorithm demonstrates superior long-term approximation to the original system compared to traditional SINDy, further validating its effectiveness.

Despite these merits, the algorithm still faces three key challenges. First, regressing the functional form of nullclines remains problematic, as overfitting is prone to occur. Although $K$-charts offer auxiliary support, they fail to guarantee the acquisition of correct functional forms in complex cases, which may distort the identified dynamic systems and undermine subsequent control performance. Second, applying SINDy to each offset in the dataset introduces additional computational overhead. This not only increases the time cost of system identification but also impairs the usability of the modified control strategy in real-time applications (e.g., robot gait control). Third, the algorithm is exclusively designed for two-dimensional dynamical systems. In high-dimensional systems, nullclines transform into hyperplanes, making it far more difficult to regress their functional forms and analyse $K$-charts, thus restricting the algorithm's scope of application.

Future research should focus on addressing the aforementioned limitations and refining technical details for nullcline-based control. For the overfitting issue, the technique in [3], which can generate the vector field and nullclines for two-dimensional dynamic systems by machine learning, may be useful to solve this problem. To reduce computational overhead, parallel computing architectures may be adopted to enhance efficiency. For high-dimensional expansion, short-term efforts could involve dimensionality reduction for high-dimensional systems (we have tried to fix certain variables to form 2D subsystems, but it doesn't work well), while long-term goals may include developing piecewise regression techniques for hyperplanes to extend the algorithm's applicability.

In summary, the optimised SINDy with nullcline reconstruction provides a valuable data-driven tool for identifying two-dimensional dynamical systems and supporting flexible control. Resolving its current limitations and enriching control-related technical details will be crucial to promoting its practical application in fields such as robot gait control and two-dimensional oscillatory system regulation.

The codes for this study are available at https://github.com/Cui-265/modified-SINDy-via-nullcline-reconstruction.

## References


[1] Paolo Arena, Luca Patané, and Angelo Giuseppe Spinosa. A nullcline-based control strategy for pwl-shaped oscillators. *Nonlinear Dynamics*, 97(2):1011–1033, 2019.

[2] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[3] Bartosz Prokop, Jimmy Billen, Nikita Frolov, and Lendert Gelens. Machine learning identifies nullclines in oscillatory dynamical systems. *arXiv preprint arXiv:2503.16240*, 2025.

[4] Bartosz Prokop, Nikita Frolov, and Lendert Gelens. Enhancing model identification with sindy via nullcline reconstruction. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(6), 2024.