Analyzing the Duffing system with Koopman theory and SINDy

Johnny Rasnic
Su Yang

MATH 645

September 4, 2023

Introduction

- In our project, we focus on the Koopman theory to investigate the 2D Duffing system (also called Duffing oscillator). Koopman theory provides a powerful way to linearize any given nonlinear dynamical system into an linear space. Such linearization process is accomplished by "Koopman eigenfunction"(see e.g. the lecture sides on chapter 5: Koopman operator). We present our progress on finding Koopman eigenfunctions of the Duffing system and also on general 2-dimensional dynamical system.
- We also observe how well the Duffing system can be obtained purely from generated data and simulated by SINDy, under various parameter ensembles, and with/without a control term during training.

Remark: Except the simulation part of this project, it focuses on autonomous system almost surely.

Linearization

- Koopman eigenfunction provides a way to linear the original nonlinear dynamics into an linear space.
- why do we bother to linearize nonlinear dynamics by Koopman eigenfunction?

Linearization(continued)

- Quick recap: Based on our previous lectures, we have studied the method to linearize any nonlinear system near its fixed points, and by classifying each fixed point in the linearized system, we can have much more insights about the behavior of the original nonlinear system by applying Hartman-Grobman theorem.
- In summary, linearization provides us with insight about nonlinear dynamics.

Koopman eigenfunction

Definition

Let's recall that Koopman eigenfunction, denoted by φ, is defined as follow,

$$K_t[\varphi](x) = \varphi(x)e^{\lambda t}, \tag{1}$$

where $\lambda \in \mathbb{R}$ is the corresponding eigenvalue of φ, and K_t is the Koopman operator.

Koopman eigenfunction

Importance

Knowing Koopman eigenfunctions of a given system guarantees analytical solutions to the original system.(why?!)

- **Pro**: This news is definitely a good one for PhD students in either engineering or physics when they are getting stuck in solving ODE by using elementary ODE solving techniques.
- **Con**: Unluckily, such eigenfunctions are hard to find for most systems.

Koopman eigenfunction

Example: Duffing system

In our project, we are interested in the autonomous Duffling system which is a 2D nonlinear system as follow,

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = 0. \tag{2}$$

We find and claim that Koopman eigenfunctions of the autonomous Duffing system (2) has the following form,

$$\varphi(x, y) = f(\frac{1}{2}y^2 + \frac{\alpha}{2}x^2 + \frac{\beta}{4}x^4)\exp(\lambda \int \frac{dx}{y(x)}) \tag{3}$$

Koopman eigenfunction

Example: Duffing system(continued)

where $y(x)$ in (3) satisfies the following equation,

$$\frac{1}{2}y(x)^2 + \frac{\alpha}{2}x^2 + \frac{\beta}{4}x^4 = c, \tag{4}$$

f is any univariate continuously differentiable function, and $c \in \mathbb{R}$ is an arbitrary constant.

Remark: It is easy to verify that (3) is indeed the eigenfunction by back substituting it into (1).

Koopman eigenfunction

General 2D system

Based on our research on the Duffing system, we find that analytical expression of Koopman eigenfunction for some special 2D nonlinear dynamics can also be constructed successfully. Such special 2D nonlinear dynamics must have the following form,

$$\dot{x} = h(x)w(y), \dot{y} = s(x)d(y), \tag{5}$$

for some real-valued, continuous functions h, w, s, d.
The following theorem is the core of this project.

Koopman eigenfunction

General 2D system(continued)

Theorem

The analytical expression of Koopman eigenfunction associated with any 2D dynamical system with a form as (5) reads,

$$\varphi(x,y) = f\big(T(y) - \int \frac{s(x)}{h(x)}dx\big)\exp\Big(\int \frac{\lambda dx}{h(x)w[T^{-1}(\int \frac{s(x)}{h(x)}dx)]}\Big), \tag{6}$$

where $T(y) = \int \frac{w(y)}{d(y)}dy$ *is assumed to be invertible, and* f *is any univariate continuously differentiable function.*

What's next?

- What happens in higher dimensions(e.g. 3D system)?
- up to this point, everything is deterministic. So, a natural question is that what if there exists randomness in the system? which is usually the case in real-life models.

Stochastic Koopman operator

Definition

The stochastic Koopman operator corresponding to a random dynamical system, denoted by φ, is defined on the set of functions $f : M \subset \mathbb{R}^n \to \mathbb{C}$, which is called the observable space, as follow,

$$K_t[f](x) = \mathbb{E}[f(\varphi(t, \omega)x)]. \tag{7}$$

Remark: Here, according to the definition of random dynamical system φ which is from the sides [1], the quantity $\varphi(t, \omega)x$ is a continuous-time stochastic process which makes the expectation in (4) meaningful.

[1] N.Crnjaric-Zic. Stochastic Koopman Operator and the Numerical Approximations of its Spectral Objects. University of Rijeka, 2019.

Corresponding eigenfunction

We are interested in finding the eigenfunctions associated with the stochastic Koopman operator which is definied in the same style as that in (1) (read the previous sides for more information in case you are interested too).
Further discussion of the stochastic Koopman theory will be delayed to Math646 as a connection of this project...

SINDy

Origins

Sparse Identification of Nonlinear Dynamics (SINDy) is a framework using spare regression methods and ideas from compressed sensing to identify governing equations for a dynamical system purely from measured data. This framework aims to optimize the number of functions and terms necessary to represent the data accurately. This framework was published by Brunton, Proctor, and Kutz in 2016. [2]

[2] Brunton, S. L., Proctor, J. L., Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences of the United States of America, 113(15), 3932–3937. https://doi.org/10.1073/pnas.1517384113

SINDy

Idea

For a given system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, we want to simulate this system into a data matrix $\mathbf{X}$ and solve the following system for Ξ:

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

where the candidate library Θ is of the form

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

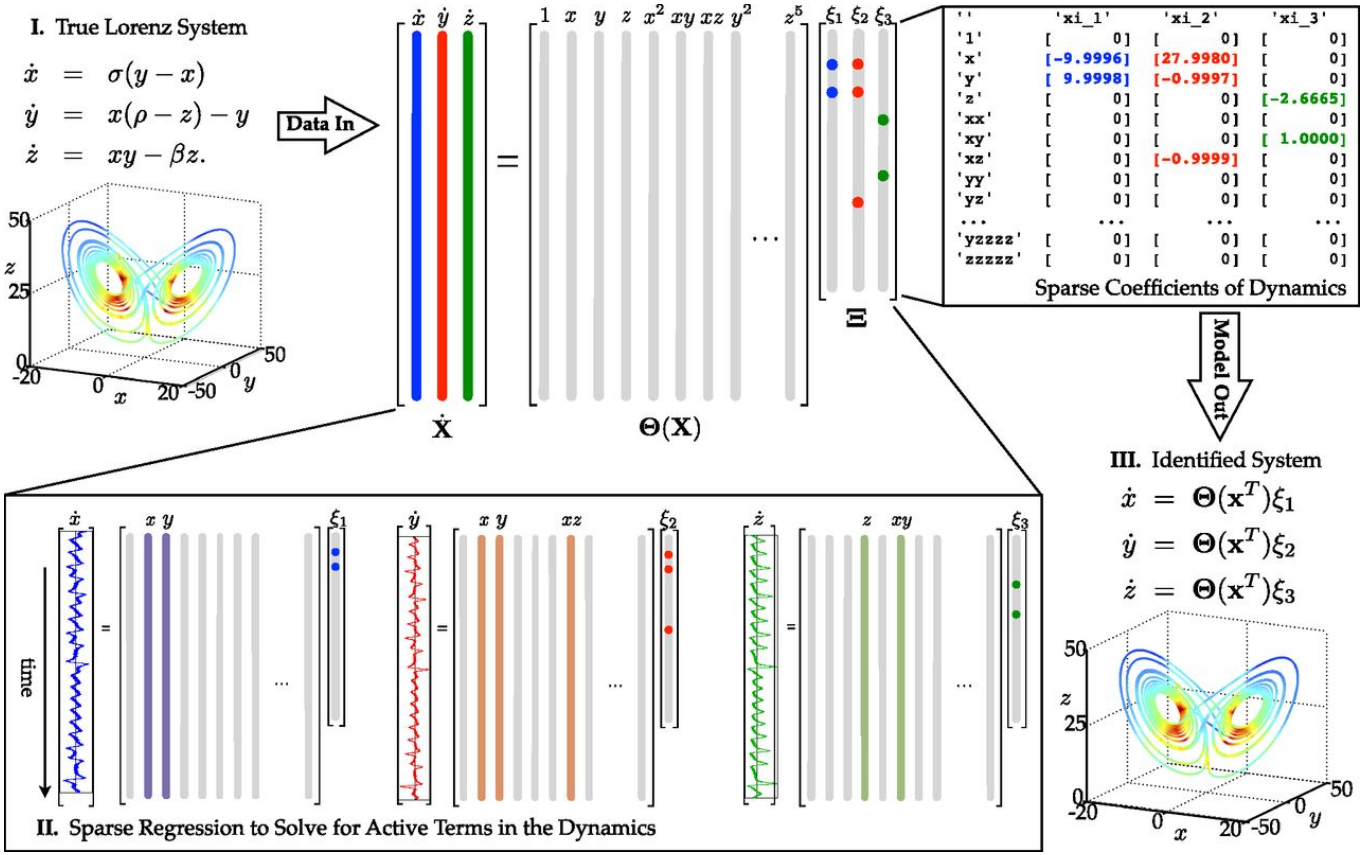
The $\mathbf{X}^{P_n}$ terms refer to the matrix of nth degree interaction terms between variables. For example $\mathbf{X}^{P_2}$ would contain entries like $x_1^2,\ x_1x_2,\ x_2x_3,\ x_2^2,\ x_3^2,$ etc. for all x_i.

SINDy

Idea

With SINDy, we are hoping to find a system to explain the data accurately and have a small number of functions used in that system. So SINDy assumes that the coefficient matrix Ξ is sparse, and uses various sparse regression methods to identify these coefficients. After training on the data, it will return a symbolic, functional system in the following form: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \Xi^T \left(\Theta \left(\mathbf{x}^T\right)\right)^T$

SINDy



3

[3] Ibid.

SINDy

A note on numerical simulation

Throughout these results, there will be comparisons between a "true simulation" and a "model simulation". The model simulation is our trained pySINDy[4] model. The true simulation is a numerical simulation of the Duffing equation with the LSODA ODE solver. Our training data will also be generated using LSODA. Under certain conditions, the Duffing equation can be quite stiff, making more common solvers like RK45 infeasible for accurate simulations. LSODA is an adaptive solver that uses an Adam's method for non-stiff portions of solving, and a BDF for stiff portions.[5]

---

[4] Silva, Brian Champion, Kathleen Quade, Markus Loiseau, Jean-Christophe Kutz, J. Brunton, Steven. (2020). PySINDy: A Python package for the Sparse Identification of Nonlinear Dynamics from Data.

[5] Hindmarsh, A C, Petzold, L R. LSODA, Ordinary Differential Equation Solver for Stiff or Non-Stiff System. NEA.

SINDy

Training

Ensemble training was used, where one numerically simulates multiple different initial conditions, and gives those to pySINDy to train on. We used a square lattice of points on $[-10, 10] \times [-10, 10]$ running for 10 time units. We then tested this trained model on new initial conditions, one inside the ensemble square, and one outside the square, for either 20 time units (for the first two results) or 100 time units (third result onward). Later, when we look at adding the control term in, we will always use the same initial condition to try and replicate some data.

SINDy

Candidate Library

For all training, We did give SINDy a candidate library containing:

- All polynomials of degree 5 (including interaction terms)
- Sine and Cosine (each with only one frequency)

SINDy

Duffing equation

The separated Duffing equation using the conditions
$\dot{x} = y, \quad \dot{z} = \omega, \quad z(0) = 0$

$$\dot{x} = y$$
$$\dot{y} = -\alpha x - \delta y - \beta x^3 + \gamma \cos(z)$$
$$\dot{z} = \omega$$

SINDy

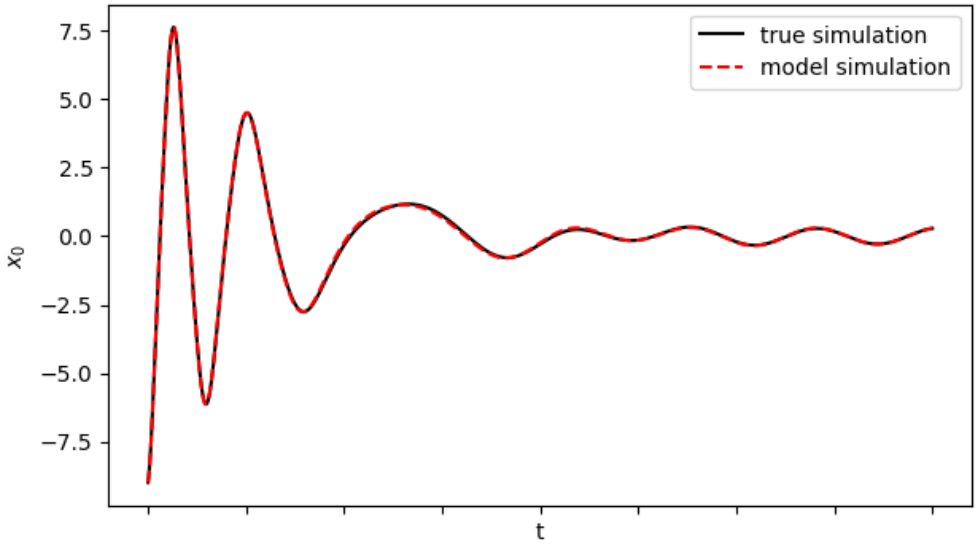
Results

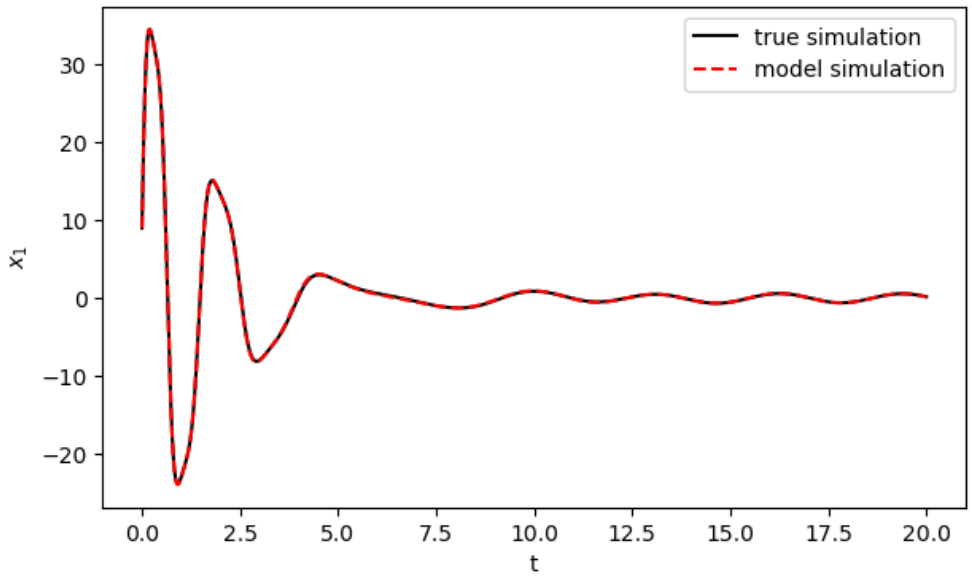
$\alpha = 1, \quad \delta = 0.8, \quad \beta = 0.4, \quad \gamma = 1, \quad \omega = 2$

```
Fitting SINDy...

Proposed system:
(x)' = 1.000 y
(y)' = -1.072 x + -0.800 y + -0.400 x^3 + 1.001 cos(1 z)
(z)' = 2.000 1
```

SINDy

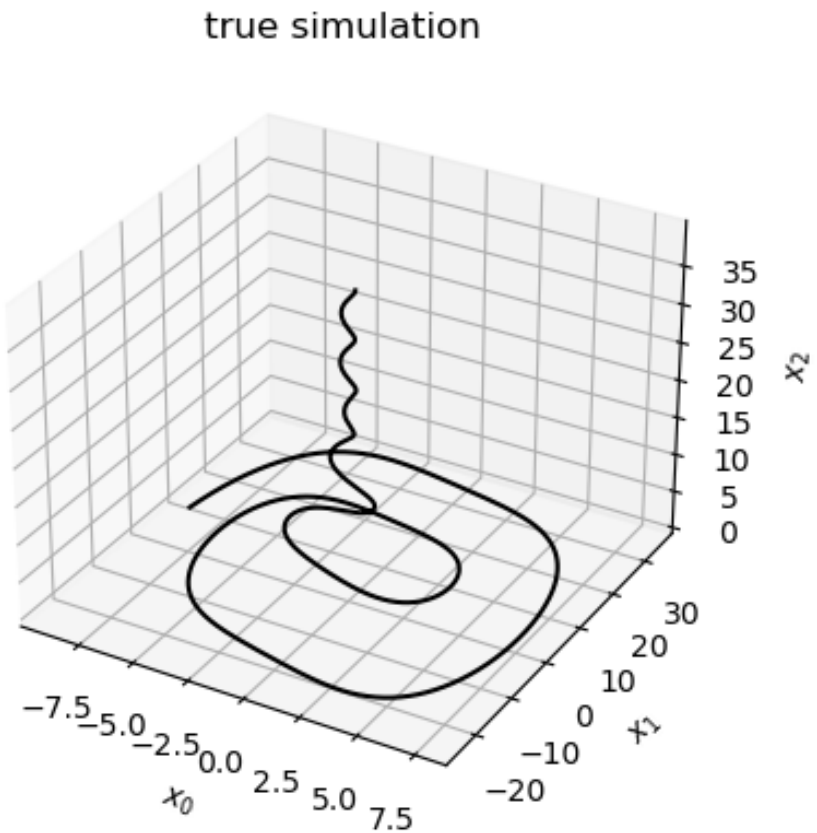
Results with Initial condition = [-9,9,0]

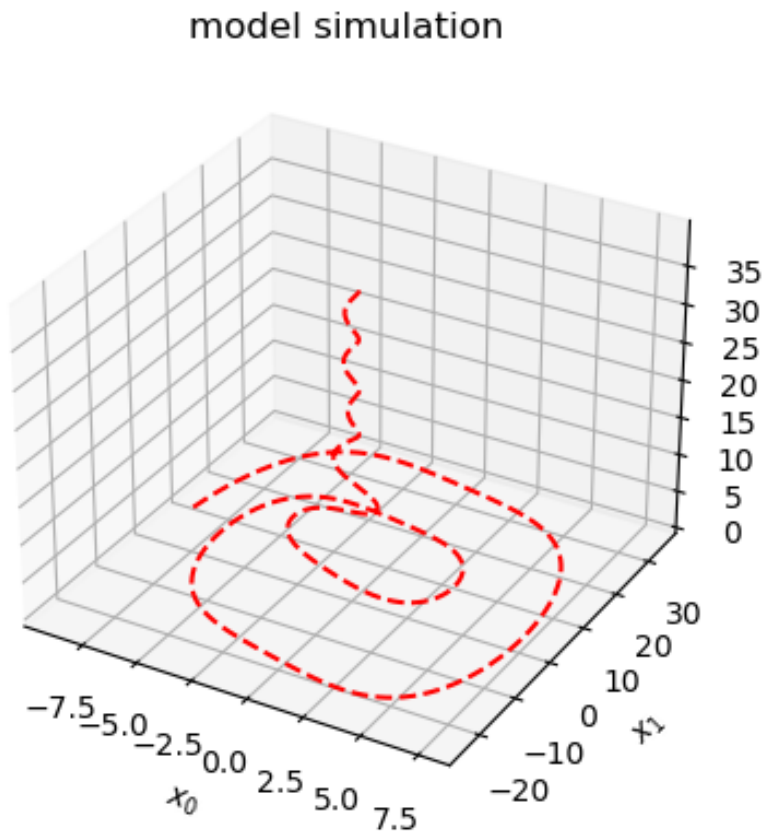




SINDy

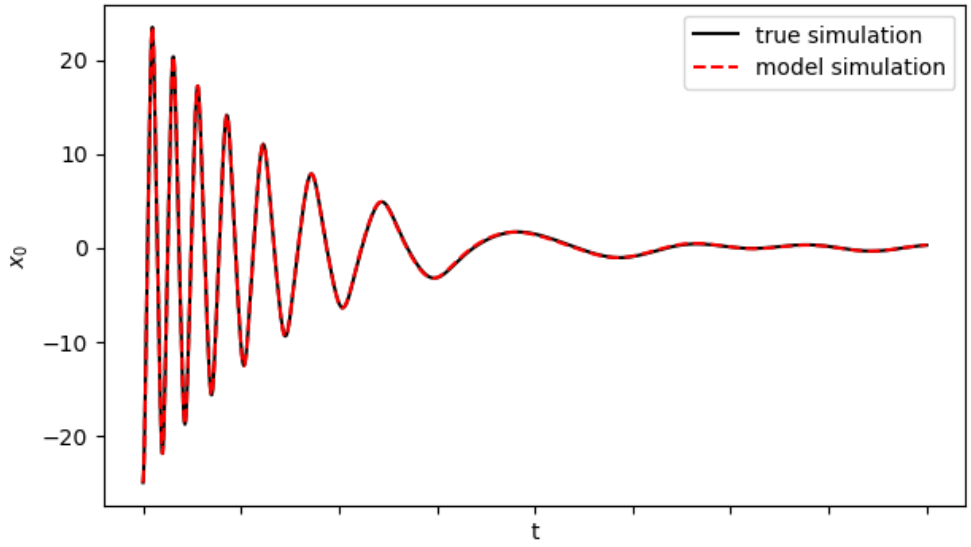
Results with Initial condition = [-9,9,0]

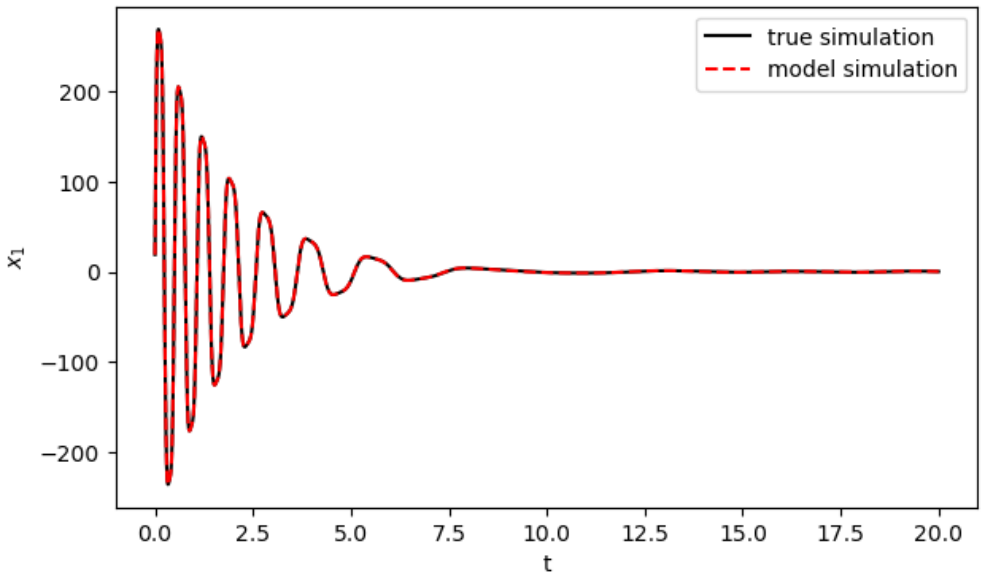




SINDy

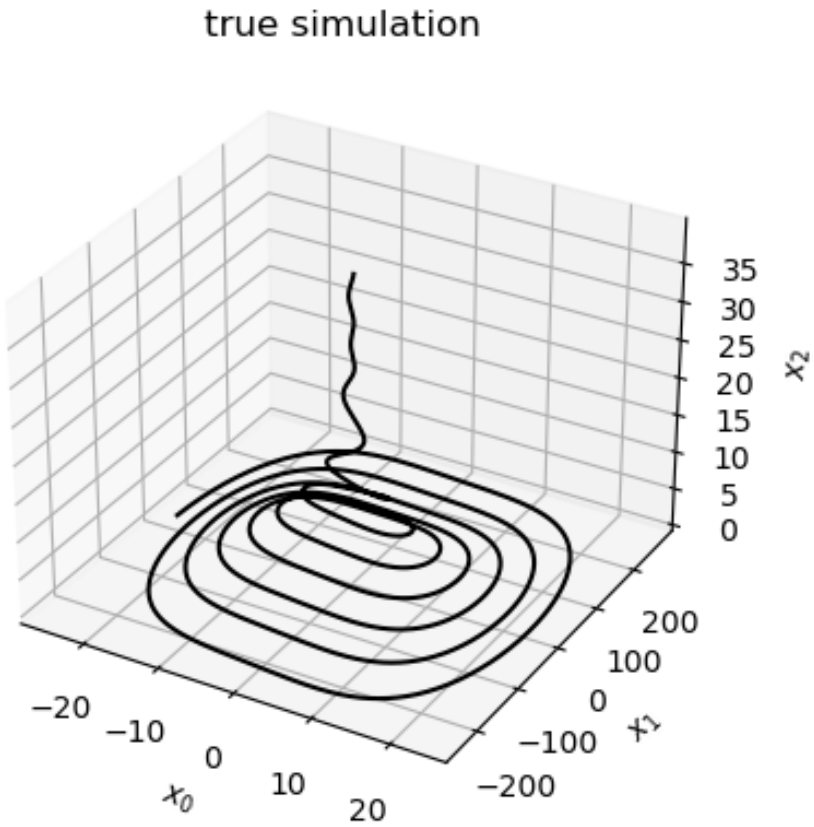
Results with initial condition = [-25,19,0]

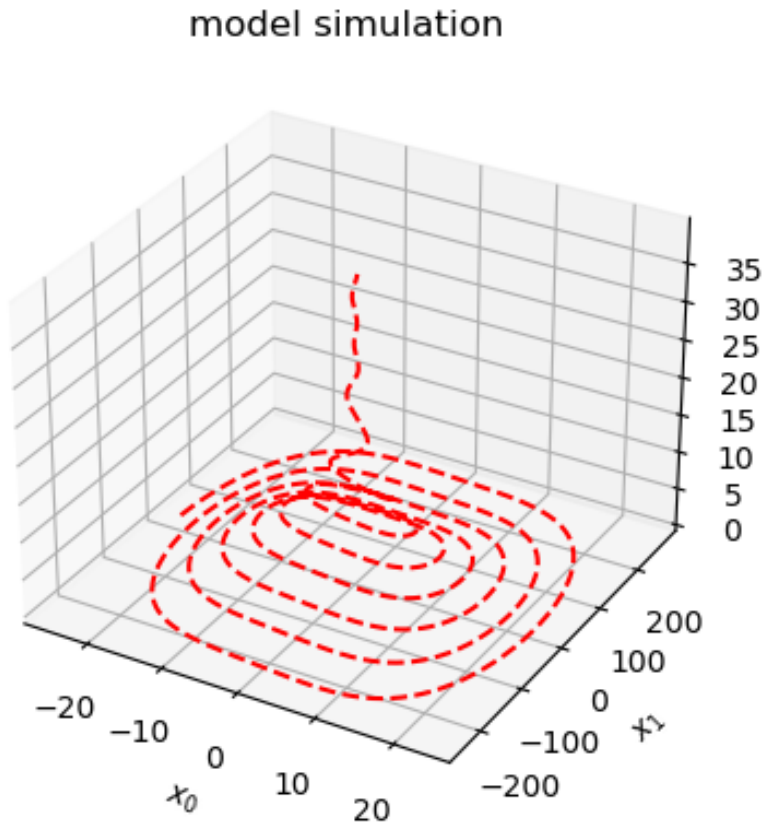




SINDy

Results with initial condition = [-25,19,0]





SINDy vs. SINDyC

Results

In many applied contexts, we can consider the forcing term $\gamma\cos(\omega t)$ as a control term, meaning we can consider the entire function as "known" while training SINDy (called SINDY with control: SINDyC[6]). For a certain set of parameters, and the initial condition $(x_0, y_0, z_0) = (1, 0, 0)$, we can vary γ and observe various cycles in the system, as well as a chaotic scenario. This parameter set is

$\alpha = -1, \quad \beta = 1, \quad \delta = 0.3, \quad \gamma$ will vary, $\quad \omega = 1.2$

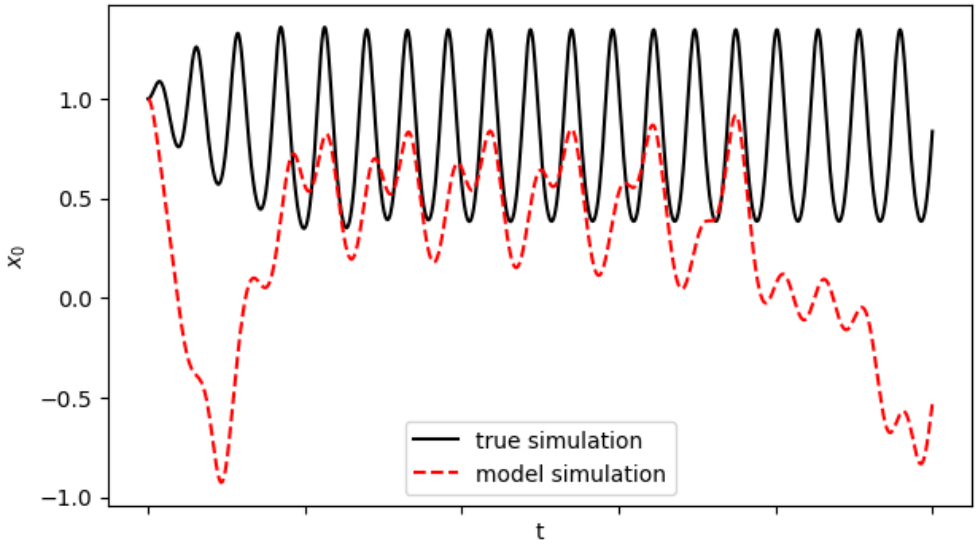
Without telling SINDy about the control term, it will have trouble identifying the α term.

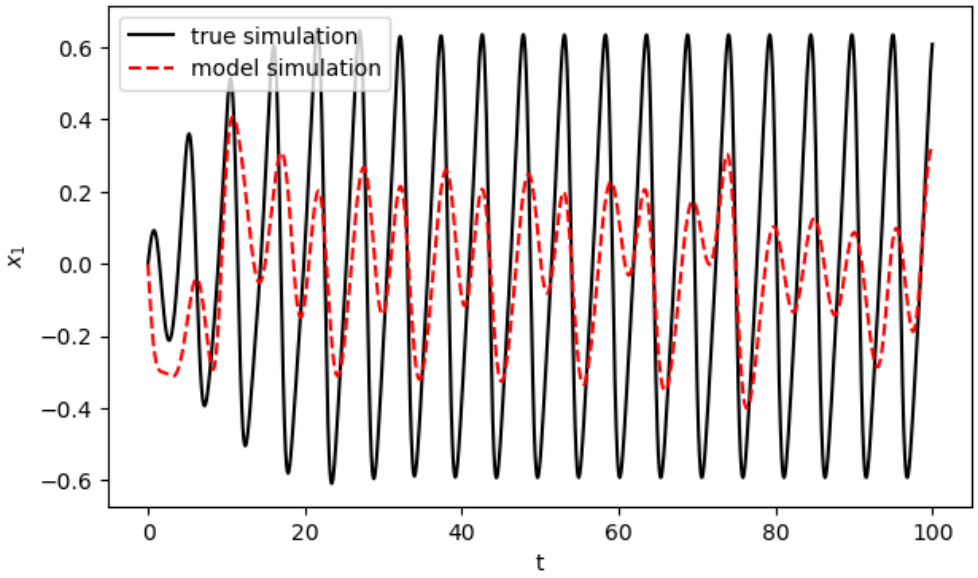
```
Fitting SINDy...

Proposed system:
(x)' = 1.000 y
(y)' = 0.393 x + -0.300 y + -0.999 x^3 + 0.200 cos(1 z)
(z)' = 1.200 1
```

[6] Brunton, S.L., Proctor, J.L., Kutz, J.N. (2016). Sparse Identification of Nonlinear Dynamics with Control (SINDYc). arXiv: Dynamical Systems.

SINDy

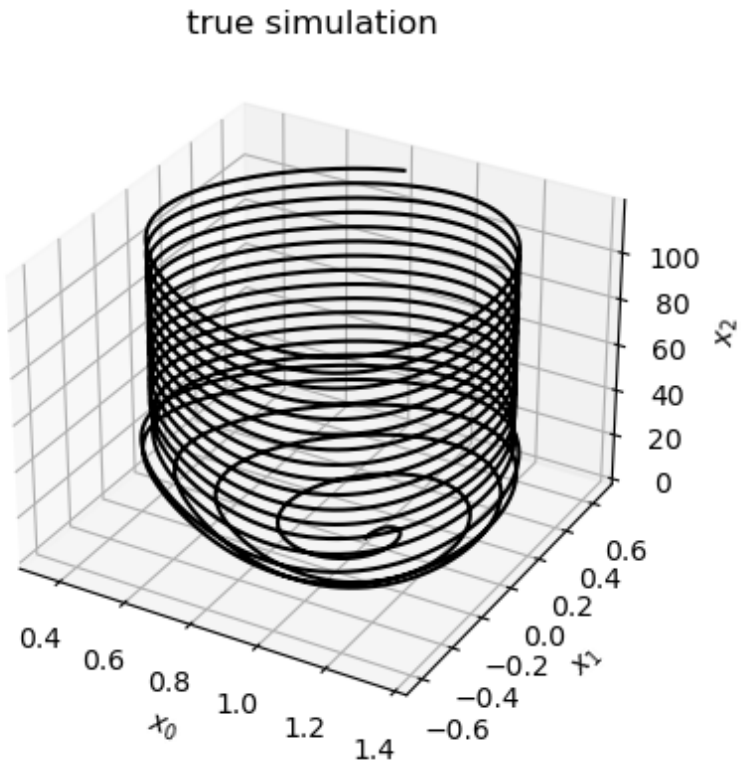
Results with initial condition = [1,0,0]

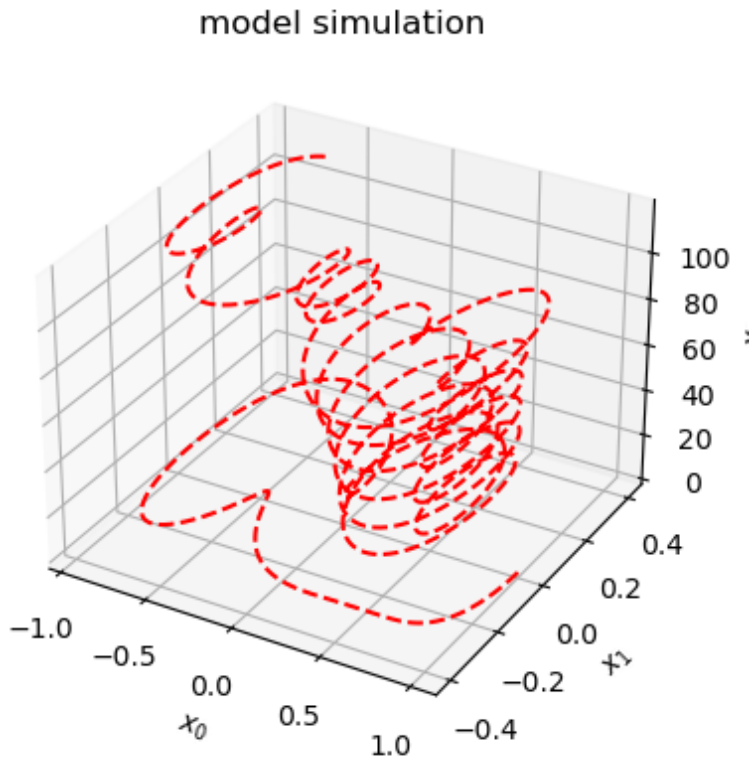




SINDy

Results





SINDyC

Idea and Results

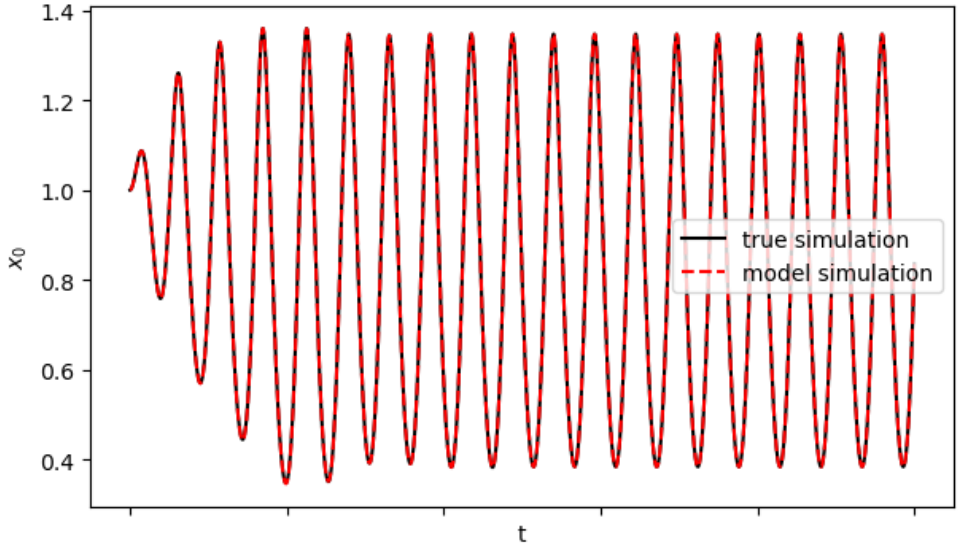
If we train SINDy with the control term, we will get much better results. The following system was identified with all the variable values of γ.

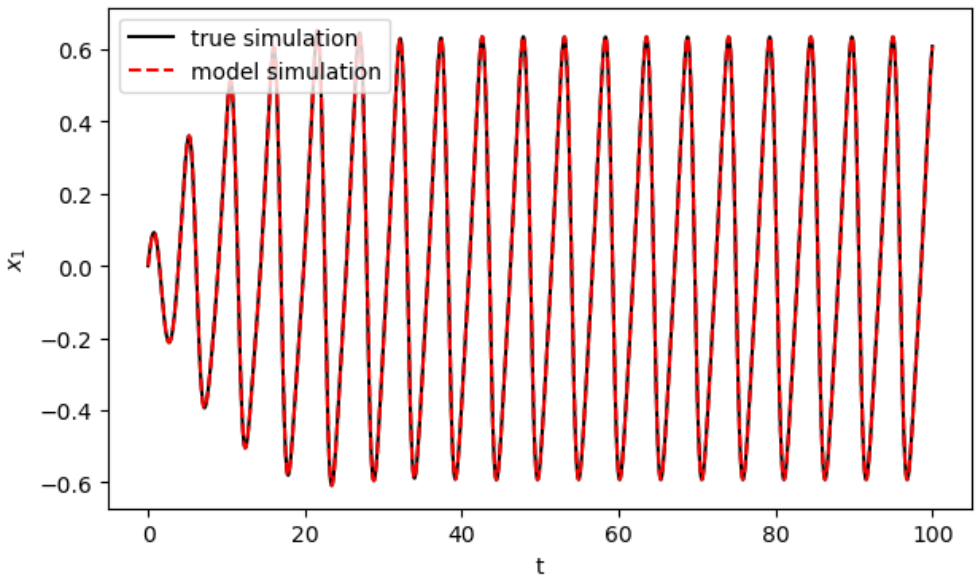
$\alpha = -1, \quad \beta = 1, \quad \delta = 0.3, \quad \gamma$ will vary, $\quad \omega = 1.2$

```
Fitting SINDy...

Proposed system:
(x)' = 1.000 y
(y)' = 0.999 x + -0.300 y + 1.000 u + -1.000 x^3
(z)' = 1.200 1
```

SINDyC

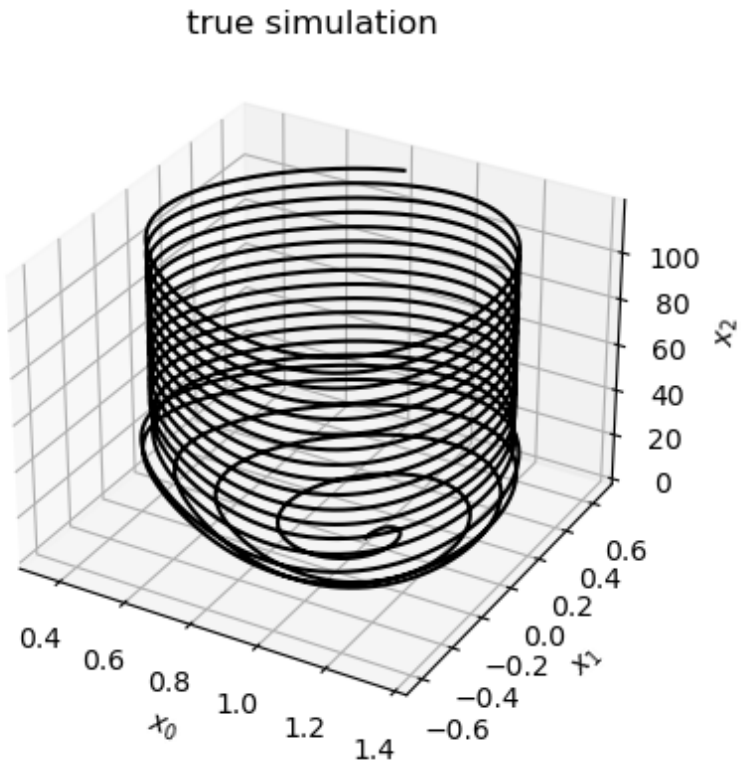
Results with $\gamma = 0.2$ (period-1)

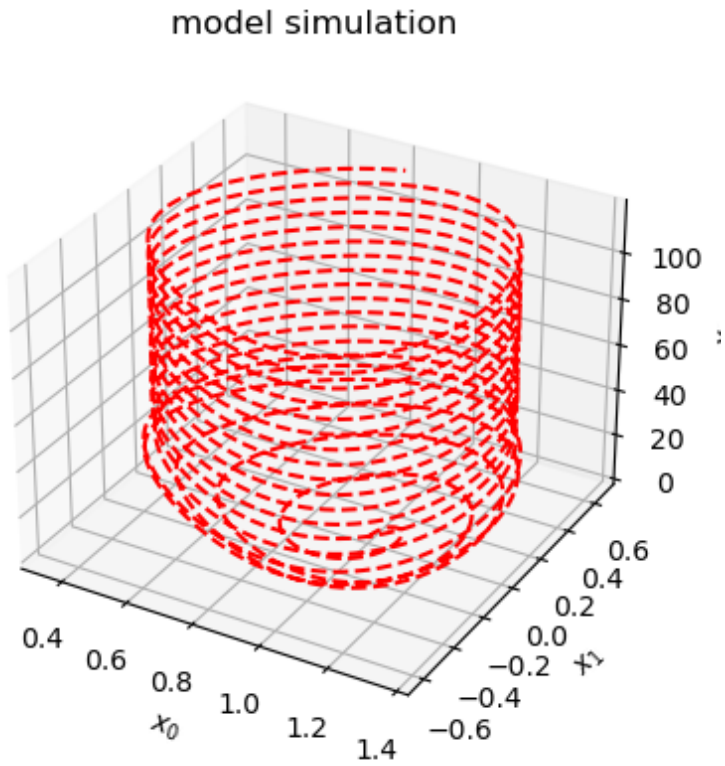




SINDyC

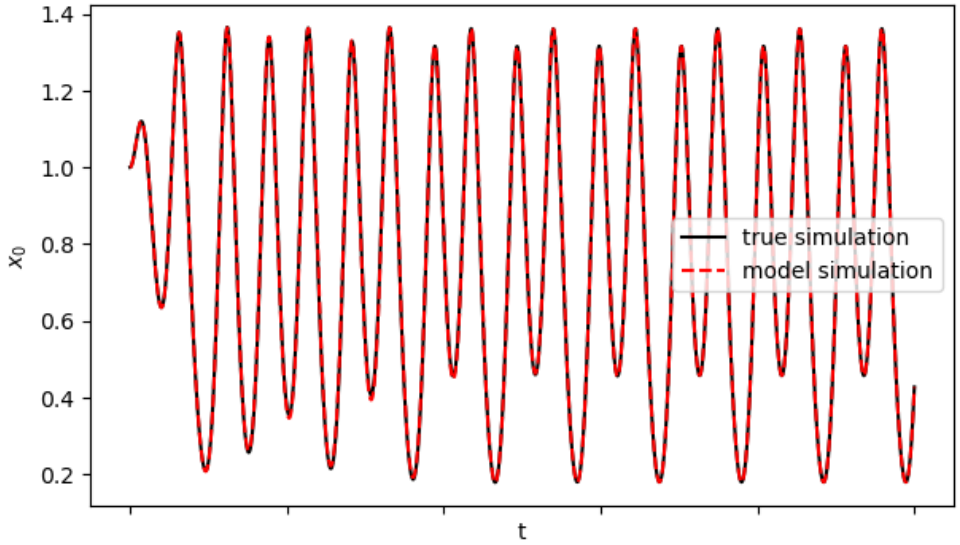
Results with $\gamma = 0.2$ (period-1)

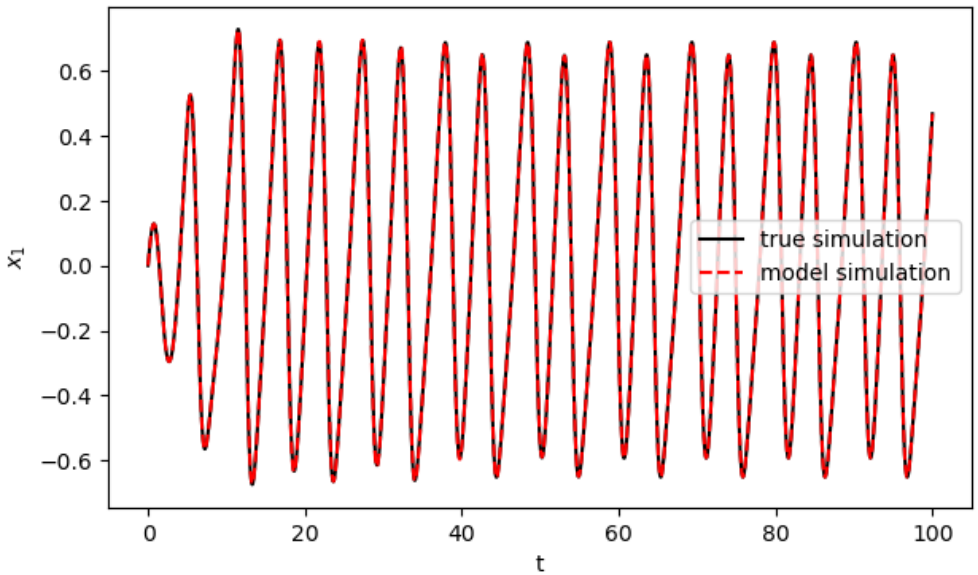




SINDyC

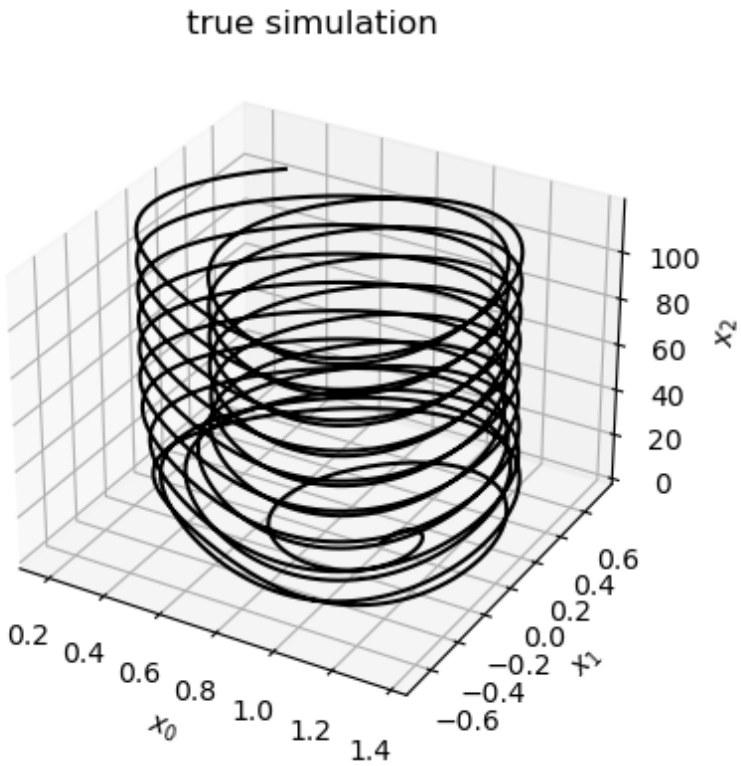
Results with $\gamma = 0.28$ (period-2)

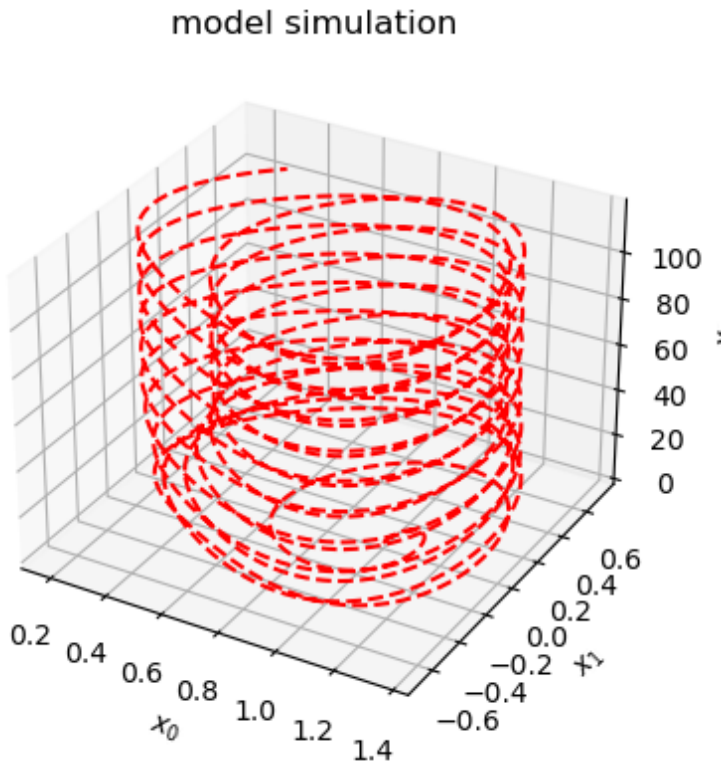




SINDyC

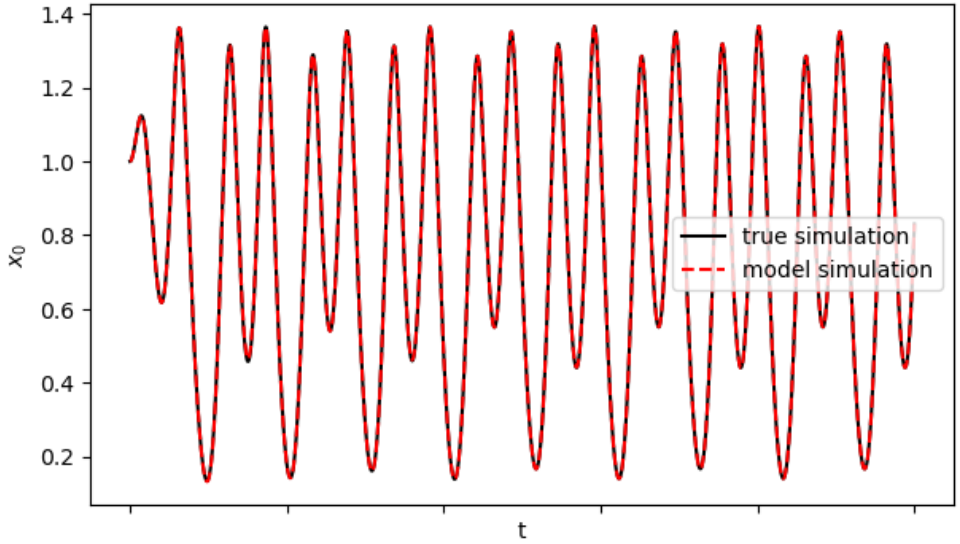
Results with $\gamma = 0.28$ (period-2)

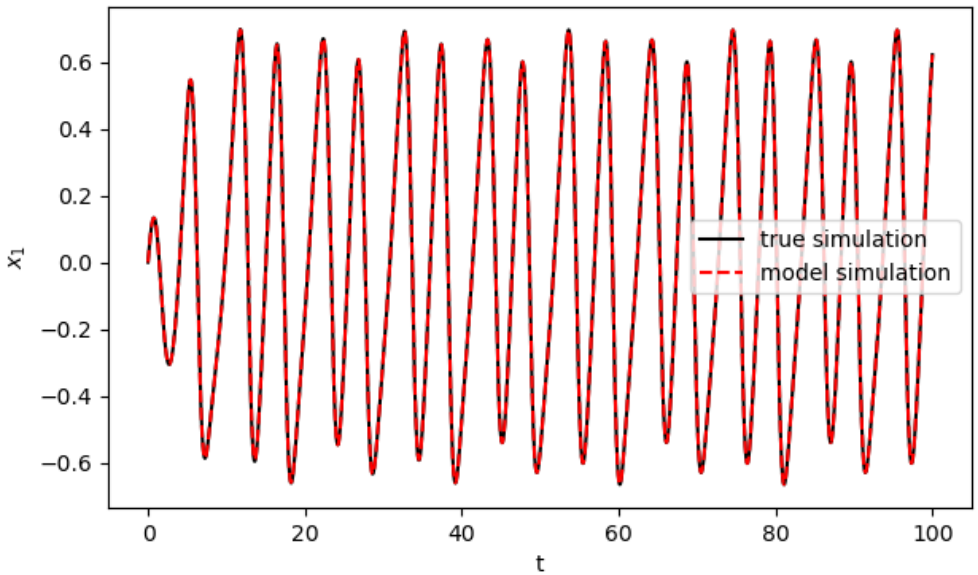




SINDyC

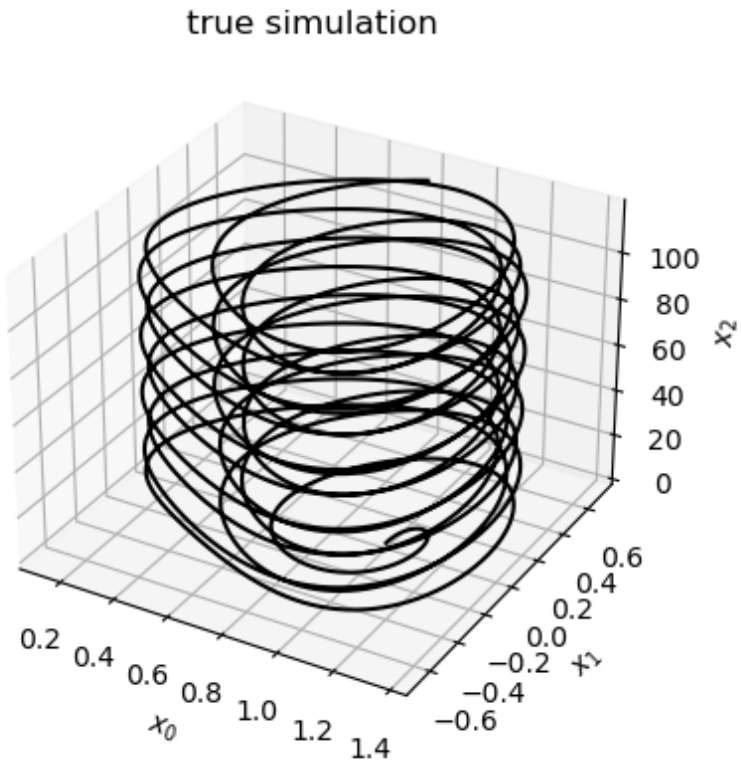
Results with $\gamma = 0.29$ (period-4)

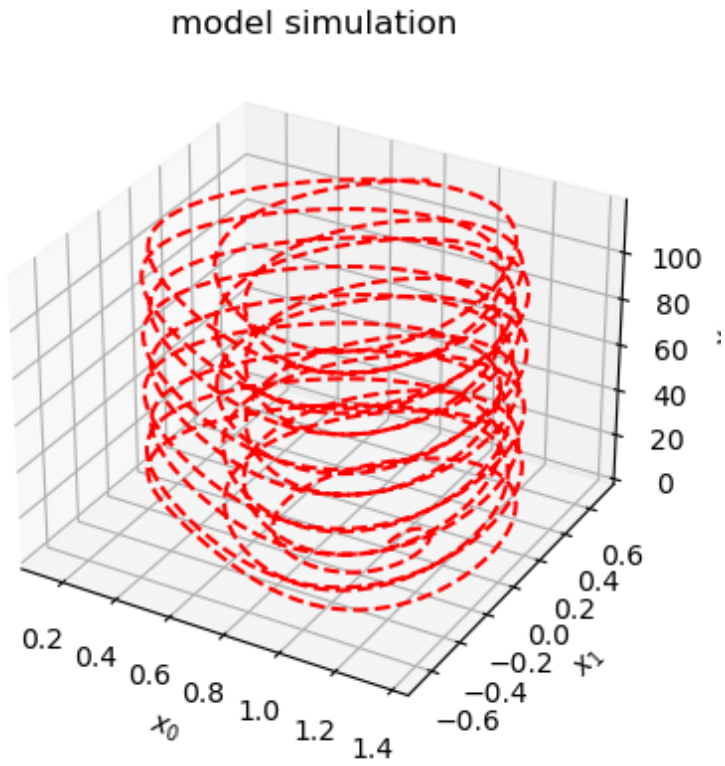




SINDyC

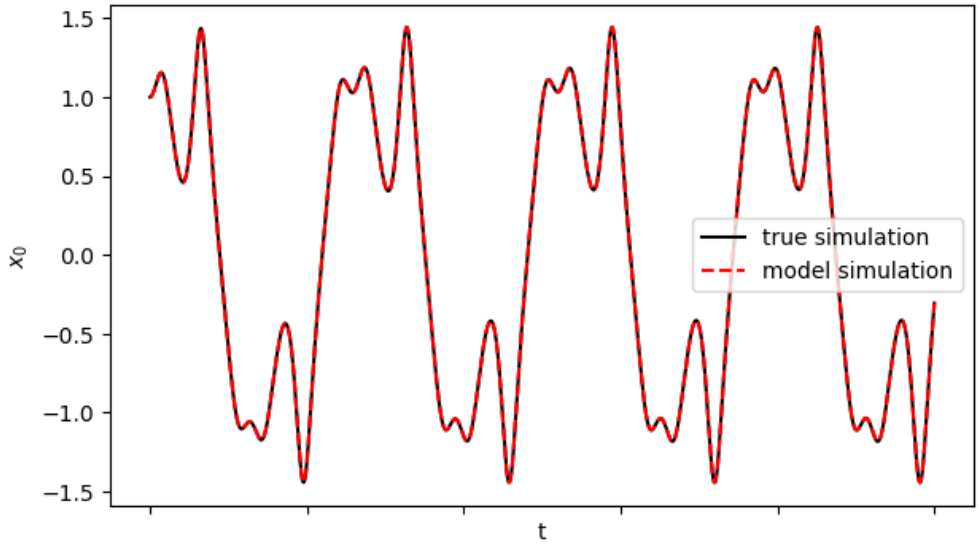
Results with $\gamma = 0.29$ (period-4)

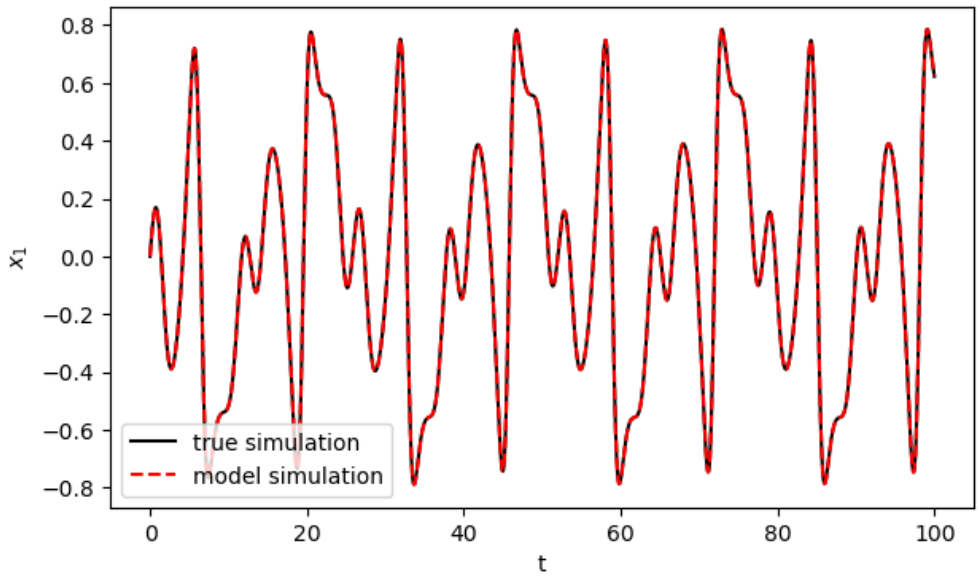




SINDyC

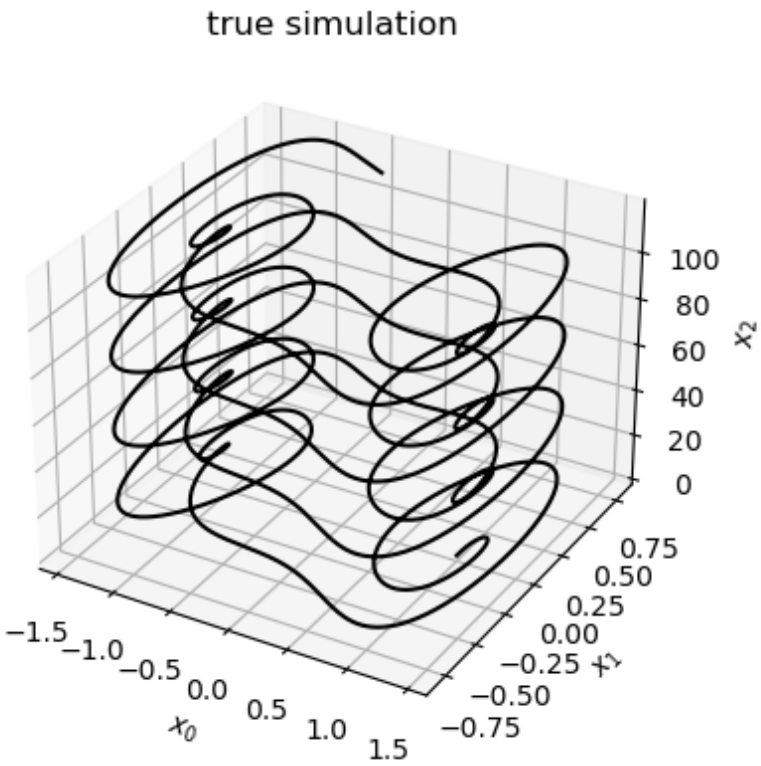
Results with $\gamma = 0.37$ (period-5)

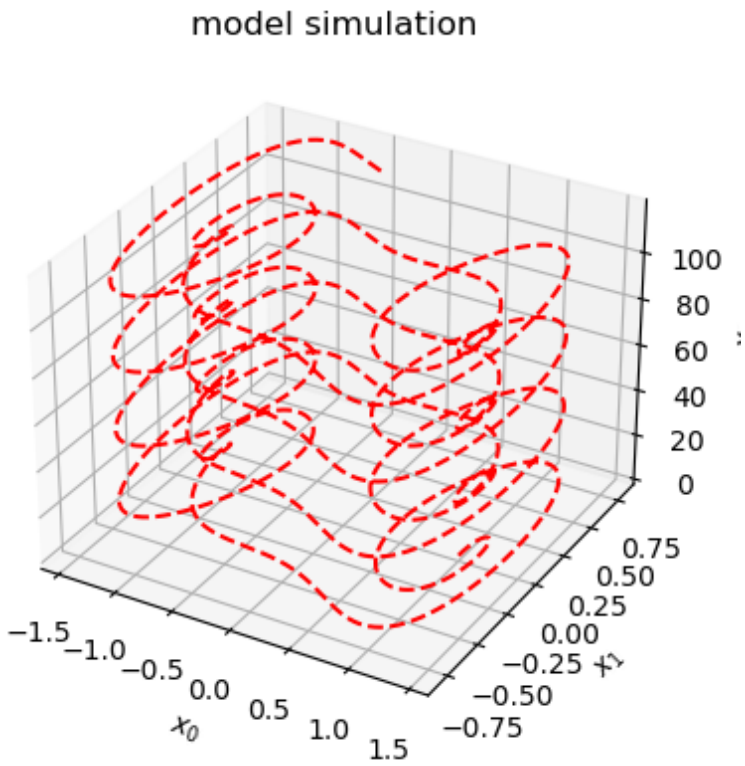




SINDyC

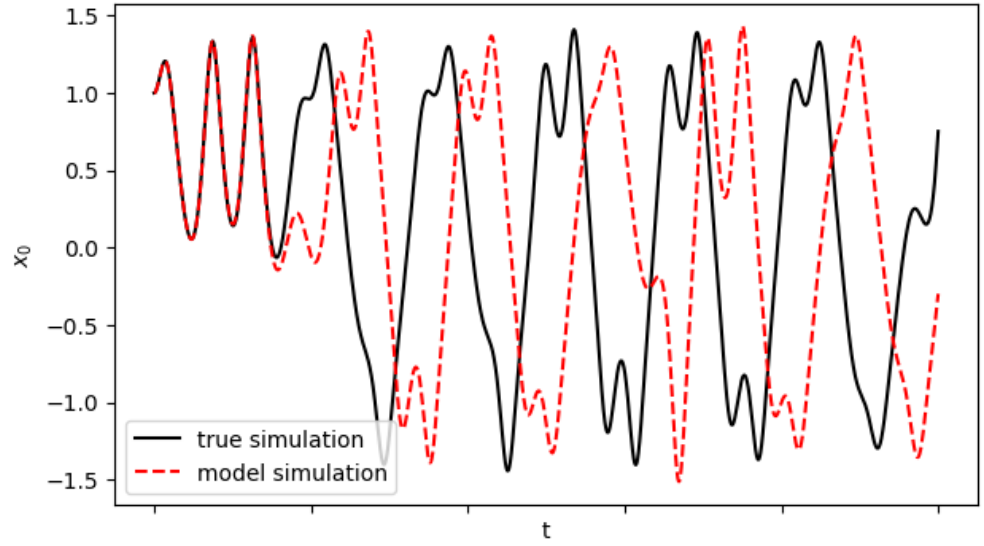
Results with $\gamma = 0.37$ (period-5)

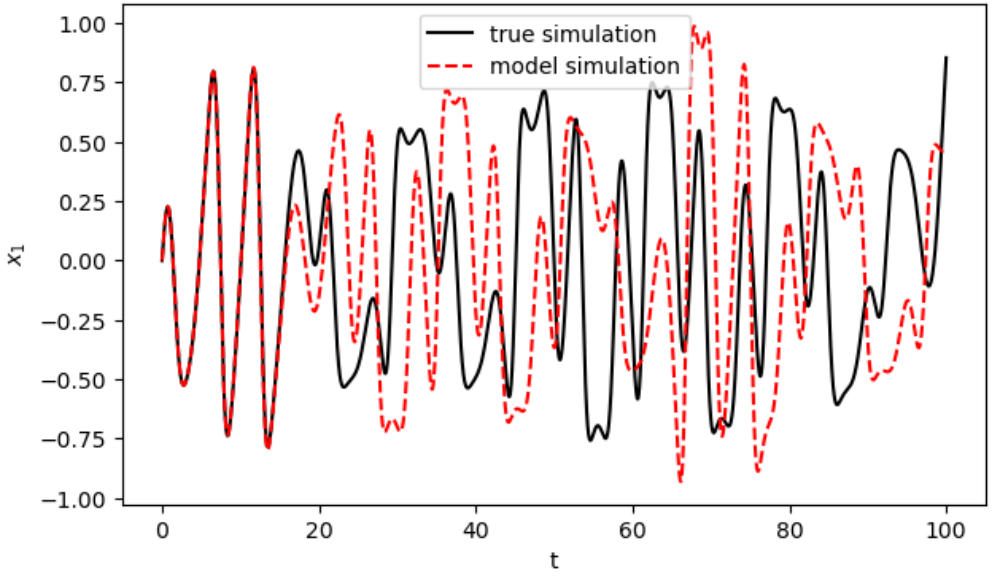




SINDyC

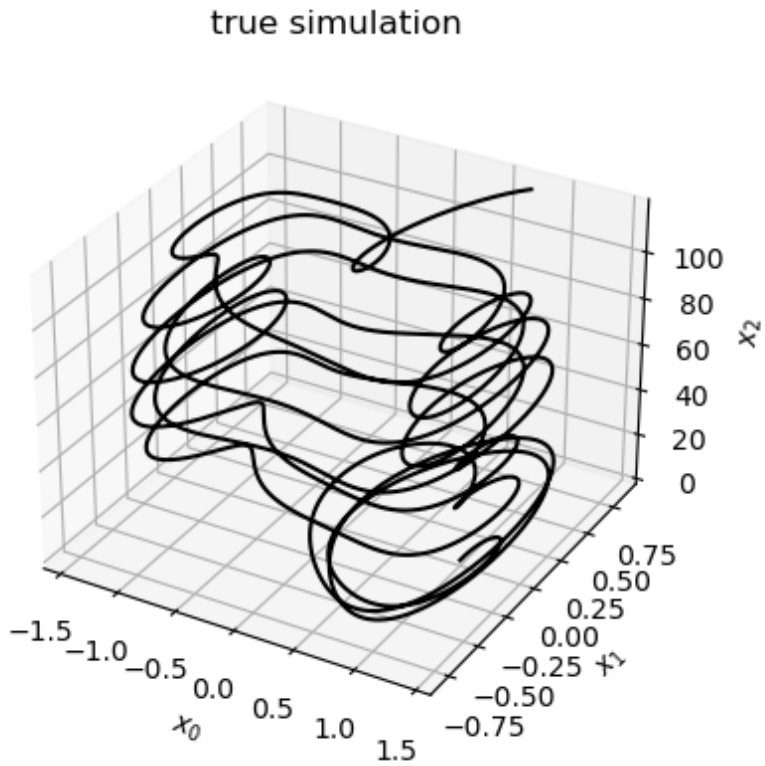
Results with $\gamma = 0.5$ (chaos)

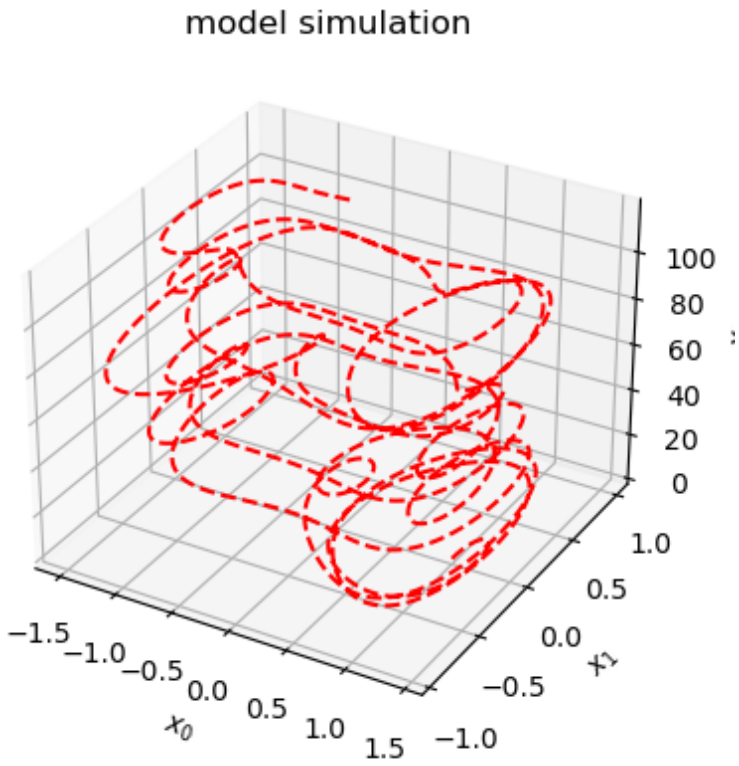




SINDyC

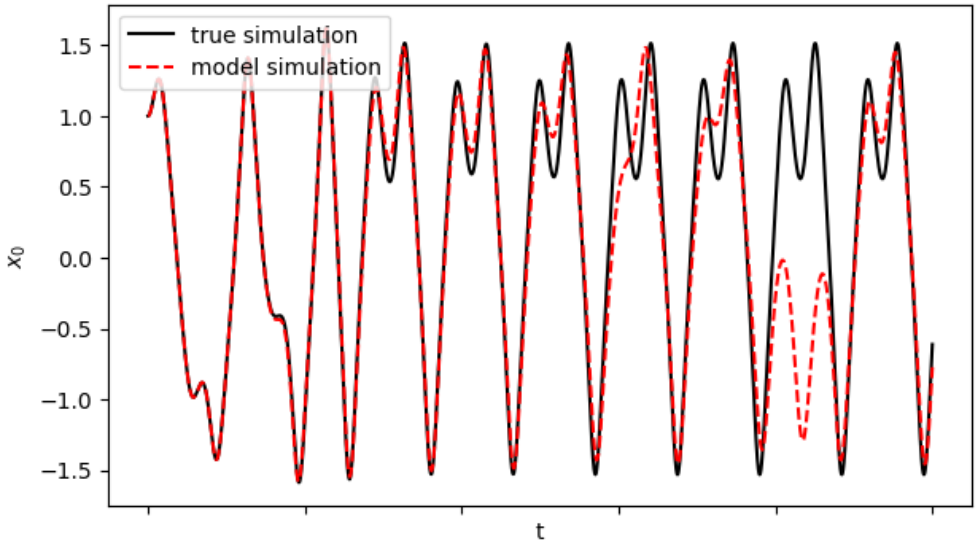
Results with $\gamma = 0.5$ (chaos)

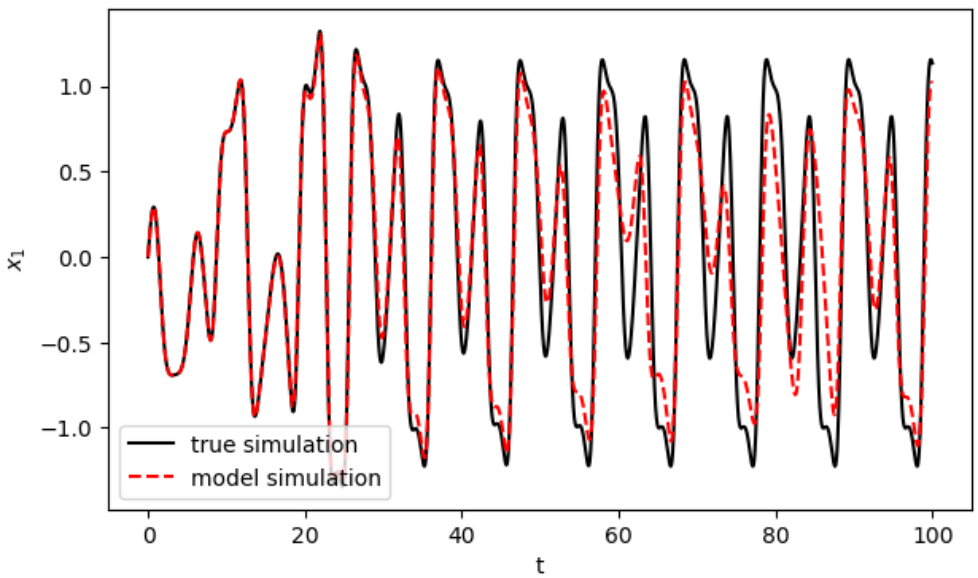




SINDyC

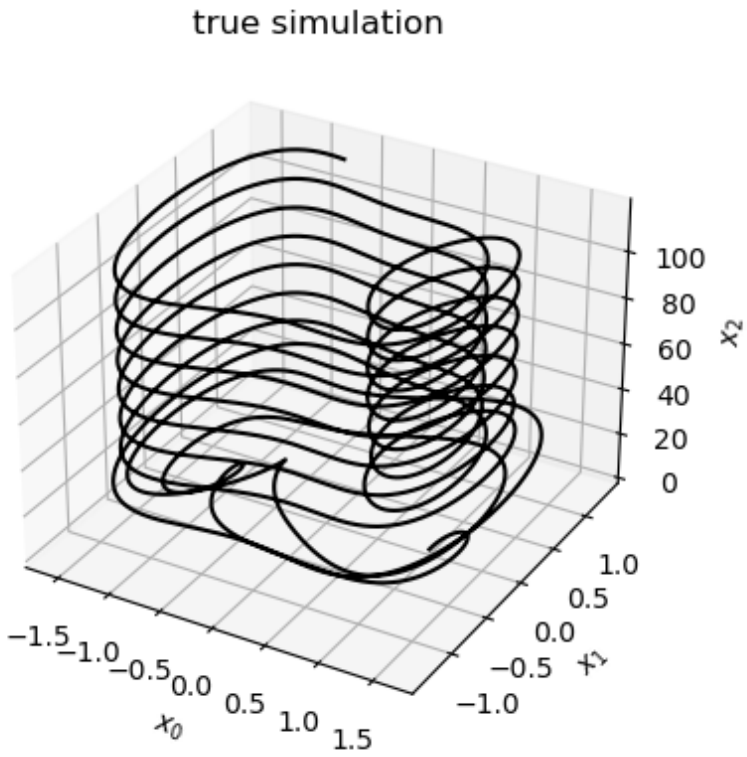
Results with $\gamma = 0.65$ (period-2)

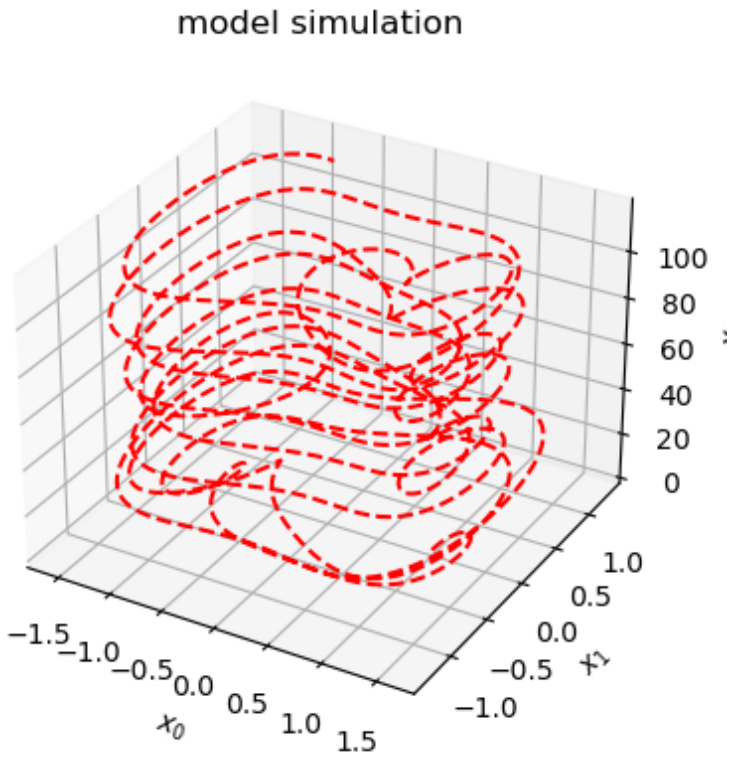




SINDyC

Results with $\gamma = 0.65$ (period-2)





Conclusion

As we can see, SINDy can recover the governing equations of the Duffing system purely from data, although it may need help in the form of training with the control term. In some cases, while the error in the model system may be small, the dynamics of the target equation may amplify this error strongly. There may also be ways around this by using more initial conditions during training, or using ensemble training, where SINDy will generate multiple different models under different regression methods and parameterizations and compares the models.