Model-Free Machine Learning for Dynamic Trajectory Tracking in Robotic Systems Using Reservoir Computing

Submitted by-

Mahavidya Satishkumar 23MDT0100

Tanmay Shinde 23MDT0113

Prathmesh Kumbhar 23MDT0122

***Guided by** - Dr. Prakash M*

Introduction

In control engineering, traditional methods for tracking control usually require a complete and accurate model of the system. But this can be challenging when the system has high dimensionality and strong nonlinearity. These systems can also be inaccurate. Because of this, there has been growing interest in developing **model-free** and **data-driven** control methods, which don't rely on exact models. Current methods often struggle to quickly adapt to changes in the system, especially when the system needs to track complex or unpredictable (chaotic) trajectories. For accurate tracking, the controller needs to respond quickly to changes in the system.

This paper addresses these issues using **machine learning**, specifically a technique called **reservoir computing**. Machine learning has changed many fields, including control systems. In the past, machine learning was used mainly for systems that work in steps, but now we can use it for more complicated systems that run continuously, which are more complex. Our method aims to track a wide range of trajectories—whether simple, complex, periodic, or chaotic—using a fully data-driven, model-free approach.

At the heart of our method is **reservoir computing**, a type of recurrent neural network (RNN) that is especially good at handling nonlinear and chaotic systems. It's very efficient because it requires less computational power and can manage complex systems. In our control system, reservoir computing is trained using only **partial observations** of the system, instead of needing the full system data.

In this study, we apply this control method to a two-arm robotic manipulator. It can track a variety of trajectories, including very complex and chaotic ones. Our system has three key features:

(1) it only requires partial observations during training and testing,

(2) it uses data from two consecutive time steps, and

(3) it uses a random input signal for training, which helps it deal with complex dynamics.

This paper presents a powerful **model-free, data-driven** control system that can track complex and chaotic trajectories, making it useful for real-world applications like robotics.

Objectives

Develop a model-free, machine learning-learning-based control framework for for precise trajectory tracking in robotic robotic systems.

Utilize reservoir computing to handle complex, dynamic trajectories with limited observational data.

Design a robust controller that can dynamically adjust to measurement noise, noise, disturbances, and system uncertainties.

Ensure accurate tracking in complex systems where traditional methods fail. fail.

Literature Review

Advances in Nonlinear and Chaotic Control: Research in controlling chaotic and nonlinear systems has developed through both traditional and data-driven approaches. Traditional methods stabilize chaotic trajectories by focusing on periodic orbits without needing full system models.

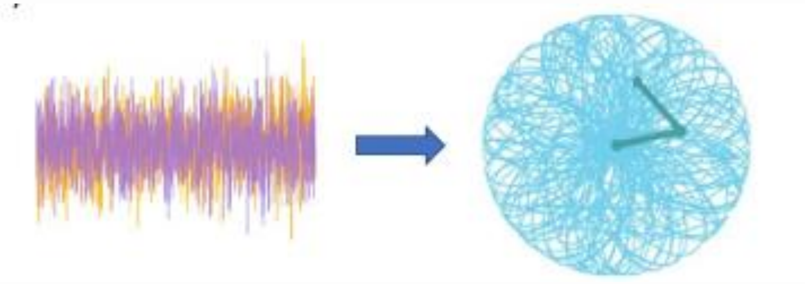
Limitations of Classical Methods: Techniques like feedback linearization, Lyapunov redesign, and sliding mode control have historically managed complex, high-dimensional systems but struggle with model inaccuracies and delays, especially in highly nonlinear environments.

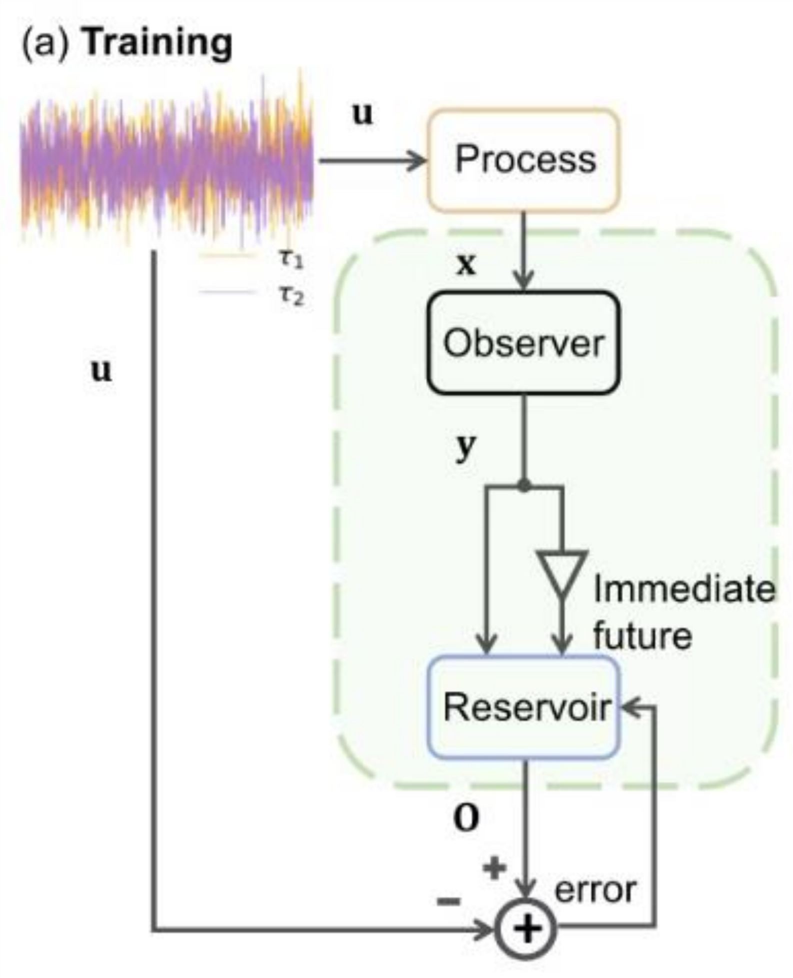
Machine Learning as a Model-Free Solution: Machine learning (ML) has introduced adaptive, model-free control methods that perform well in situations without explicit system models. Reinforcement learning and reservoir computing offer flexible and adaptive control strategies for dynamic environments.

Effectiveness of Reservoir Computing: Reservoir computing stands out for its efficiency and real-time adaptability in nonlinear control, making it suitable for tracking chaotic dynamics in high-dimensional systems. It supports complex trajectory tracking in robotics and other domains where dynamic control is essential.

Methodology

Training Phase





- The control input generated using stochastic signals is $u(t) = [\tau_1(t), \tau_2(t)]^T$.
- The full system state is represented by an eight-dimensional vector:

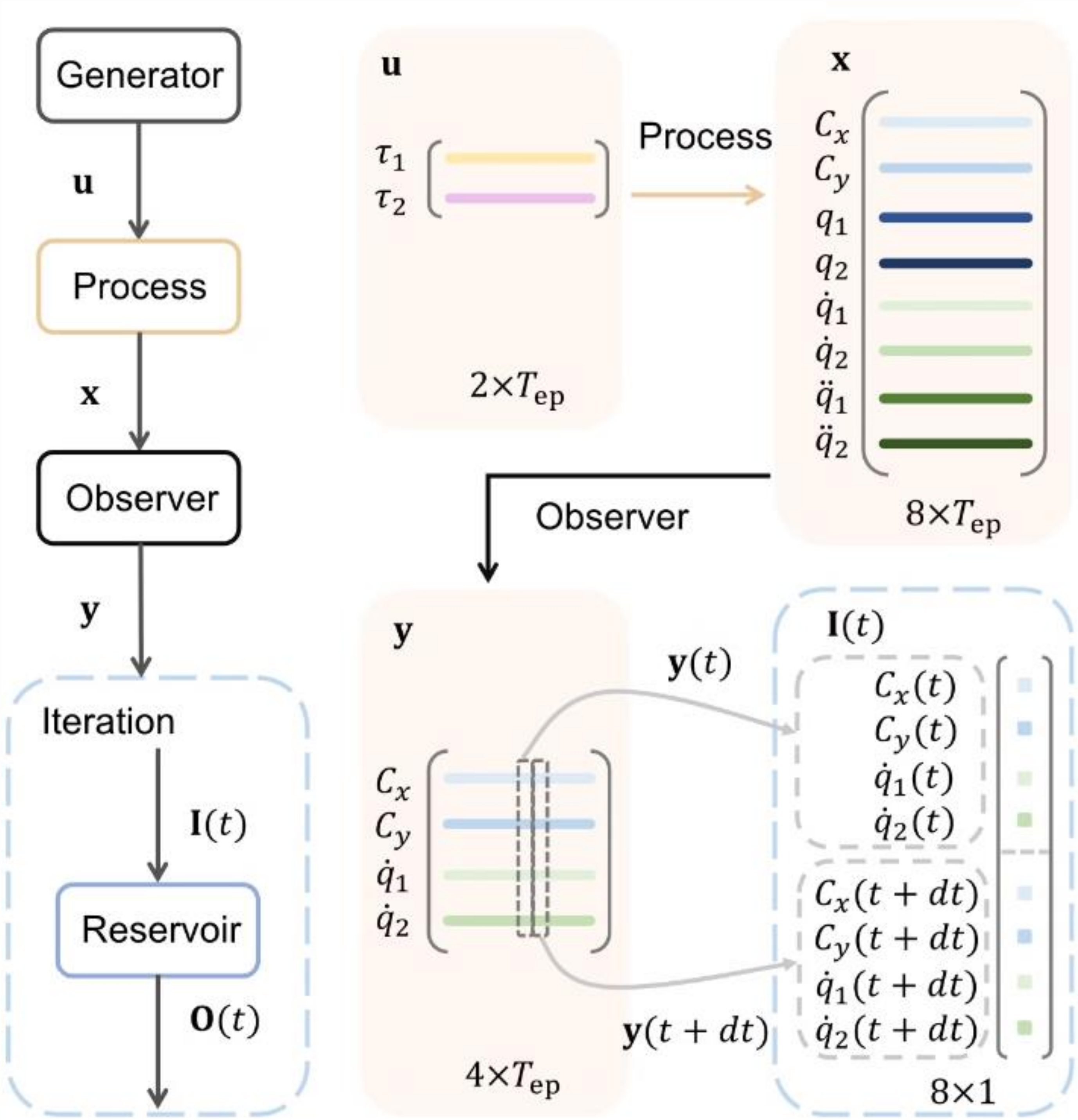
$$x(t) = [C_X, C_Y, q_1, q_2, \dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2]^T$$

- C_X and C_Y are the Cartesian coordinates of the end effector.
- q_i represents the angular position.
- $\dot{q}_i$ represents the angular velocity.
- $\ddot{q}_i$ represents the angular acceleration of arm i (where i =1,2).
- The observed states are captured in a four-dimensional vector is,

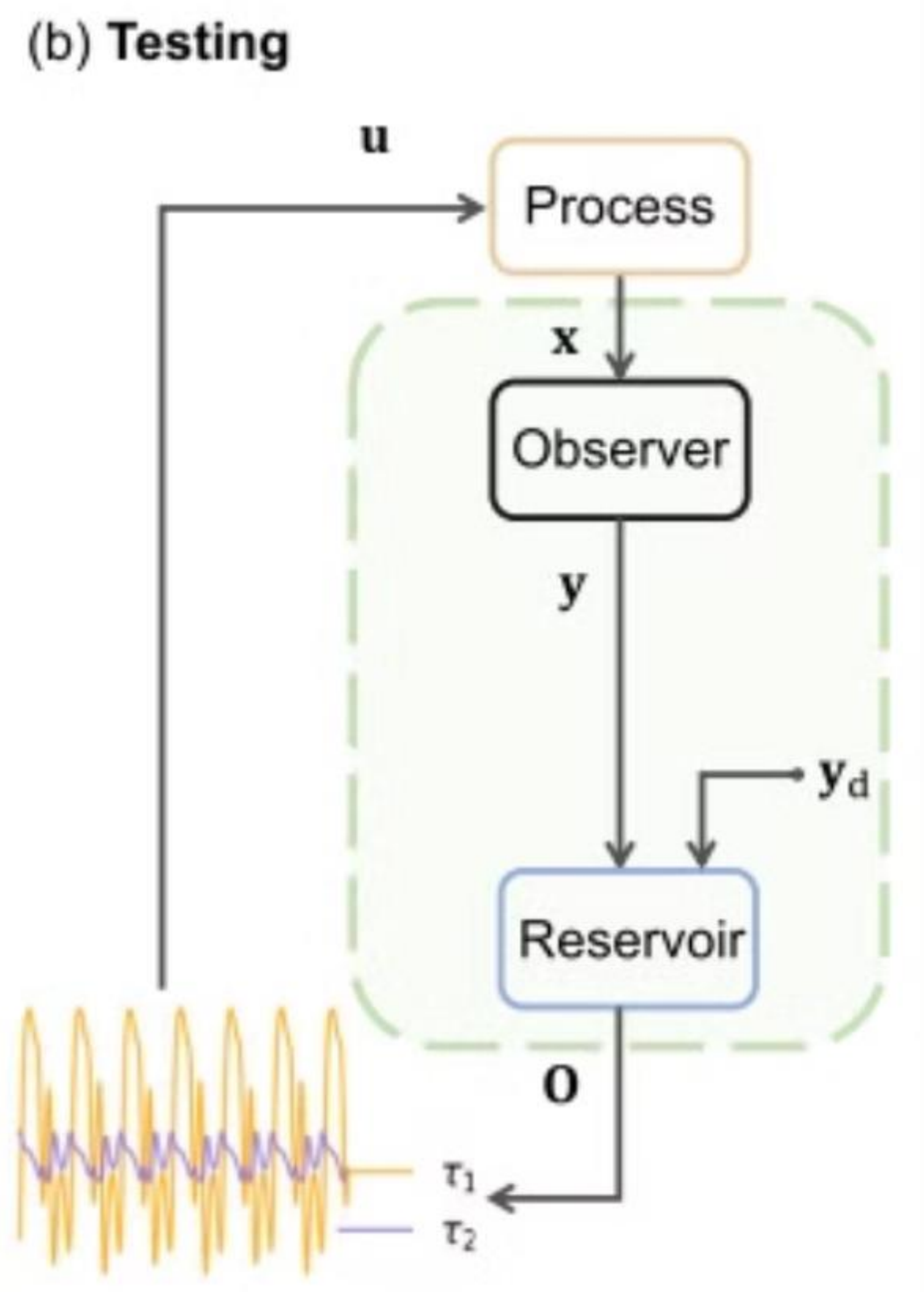
$$\mathrm{y}(t) = [C_X, C_Y, \dot{q}_1, \dot{q}_2]^T$$

- The loss function is defined as $\mathrm{e}(t) = u(t) - O(t)$. where $O(t)$ is the predicted control signal, which helps guide the model's learning.

Data Structure and Variable Flow in Training Phase of Machine-Learning Controller

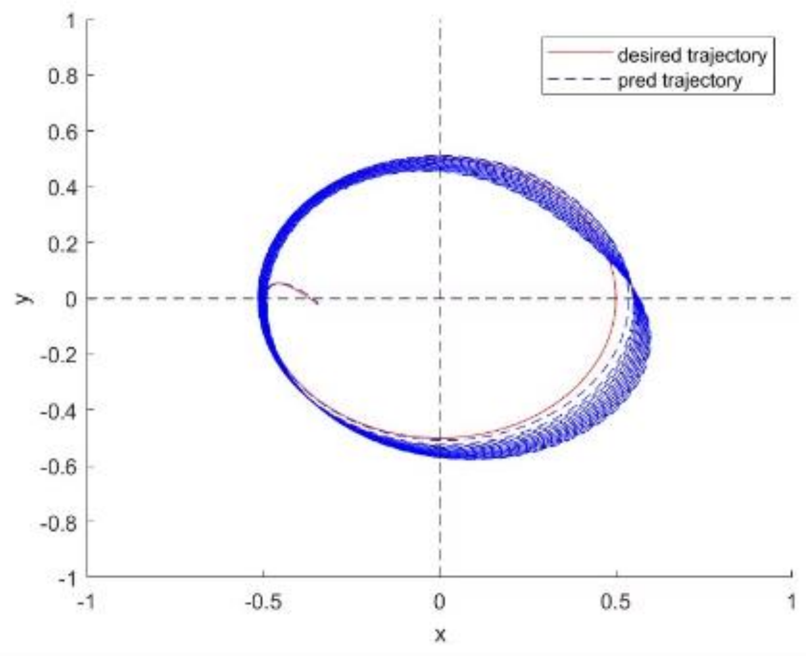


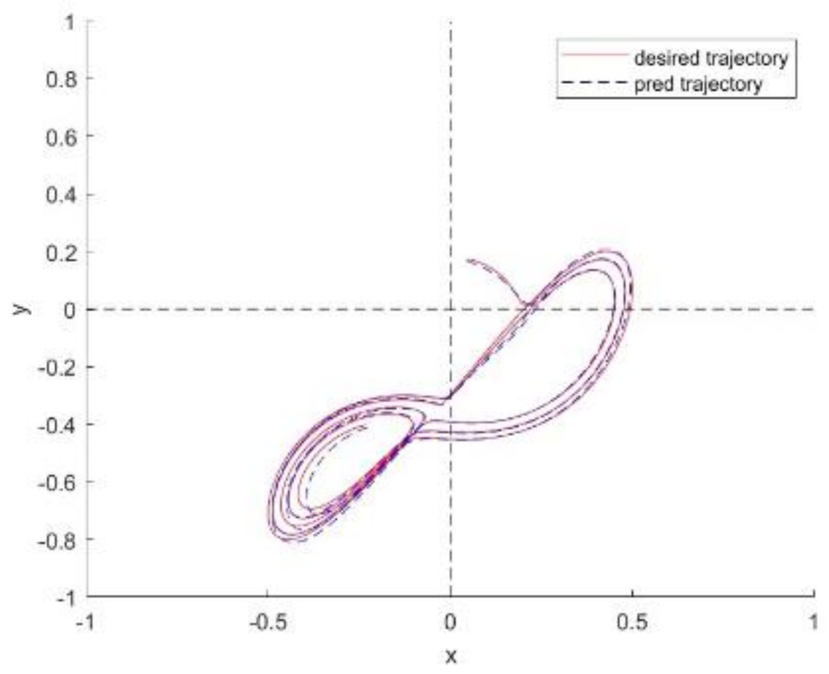
Testing and Deployment Phase



- The neural network inverts system dynamics to determine necessary control signals.
- It calculates control signals based on the current $y(t)$ and desired future state $y_d(t + dt)$.
- Adjusts output to minimize error between $y(t + dt)$ and $y_d(t + dt)$.
- Can accurately track both simple and chaotic trajectories.
- Utilizes a reservoir controller to map current to desired positions for precise control.
- Tested on 16 complex trajectories, including chaotic paths.

Results





Circular reference trajectory

Described as: $x = a.\cos(\frac{2\pi t}{T_p})$

$$y = a.\,sin(\frac{2\pi t}{T_p})$$

where, size of trajectory a = 0.5

period of trajectory T_p= 150

Chaotic Lorenz reference trajectory

Dataset contains the values of x(t), y(t) and z(t) with fixed parameters.

Some of the other reference trajectories

Chaotic Mackey-Glass reference trajectory

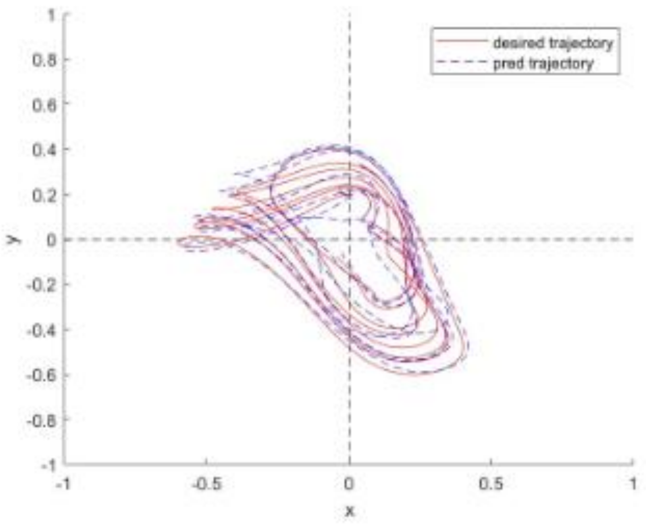
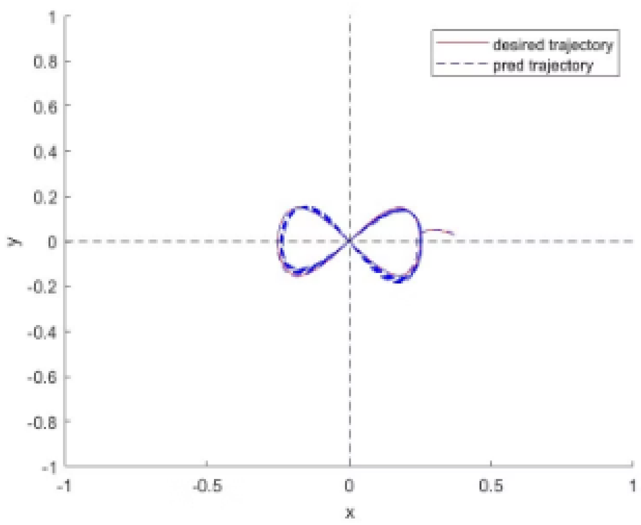
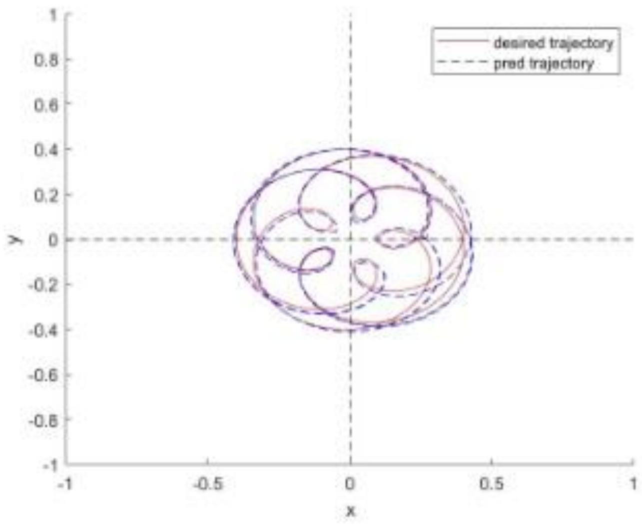


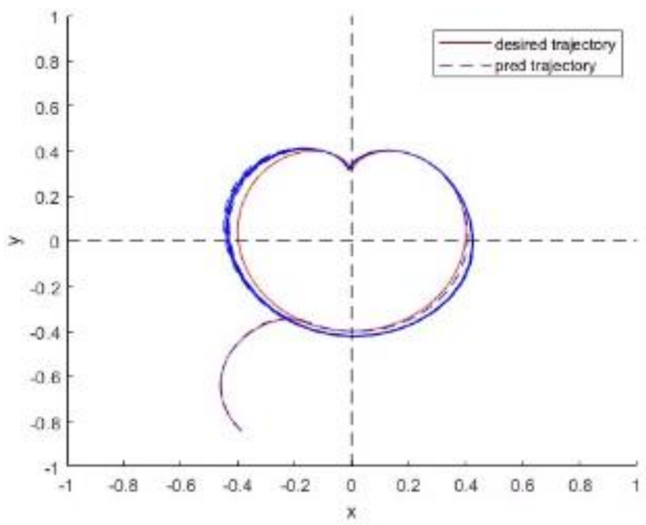
Figure-8 reference trajectory



Epitrochoid reference trajectory



Heart-shaped reference trajectory



Conclusion

- Achieves model-free, real-time tracking for complex and chaotic trajectories using a machine-learning-based approach, eliminating the need for a detailed system model.
- Requires only a single training session to handle varied trajectories, unlike traditional controllers that need custom tuning, making it practical for diverse applications.
- Enables adaptive and precise control for robotics, autonomous vehicles and medical devices, especially in dynamic and unpredictable environments.
- Effective for autonomous navigation, laser cutting, 3D printing, and agricultural robots, where adaptability and trajectory tracking are critical.
- Promising for complex research domains, including chaotic system control, climate modelling, and other fields where adaptive, model-free control offers significant advantages.

References

- Zhai, Zheng-Meng, et al. "Model-free tracking control of complex dynamical trajectories with machine learning." Nature communications 14.1 (2023): 5698.
- Zhai, Z. -M. Chaotic trajectories. Zenodo https://doi.org/10.5281/ zenodo.8044994 (2023).
- Grebogi, C. & Lai, Y.-C. Controlling chaotic dynamical systems. Sys. Cont. Lett. 31, 307–312 (1997).
- Grebogi, C. & Lai, Y.-C. Controlling chaos in high dimensions. IEEE Trans. Cir. Sys. 44, 971–975 (1997).
- Boccaletti, S., Grebogi, C., Lai, Y.-C., Mancini, H. & Maza, D. Control of chaos: theory and applications. Phys. Rep. 329, 103–197 (2000).
- Zañudo, J. G. T., Yang, G. & Albert, R. Structure-based control of complex networks with nonlinear dynamics. Proc. Natl Acad. Sci. USA 114, 7234–7239 (2017).
- Klickstein, I., Shirin, A. & Sorrentino, F. Locally optimal control of complex networks. Phys. Rev. Lett. 119,268301(2017).

Thank You