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

Tanmay Arun Shinde
Dept. of Mathematics
Vellore Institute of Technology
Vellore, India
tanmayarun.shinde2023@vitstudent.ac.in

Mahavidya Satishkumar
Dept. of Mathematics
Vellore Institute of Technology
Vellore, India
mahavidya.2023@vitstudent.ac.in

Prathmesh Kumbhar
Dept. of Mathematics
Vellore Institute of Technology
Vellore, India
kumbharprathmesh.2023@vitstudent.ac.in

Dr. Prakash M
Dept. of Mathematics
Vellore Institute of Technology
Vellore, India
prakash.m@vit.ac.in

***Abstract*— Achieving nonlinear tracking control, essential in robotics for following intricate motion paths, generally requires comprehensive system modeling and equations. This study proposes a machine learning model-free approach to control a robotic two-arm manipulator using limited observable states. The framework employs reservoir computing, utilizing stochastic input data to train on partial state observations at two consecutive time steps. During testing, the trained model aligns the system with a target trajectory by replacing the future state with the desired trajectory state. We validate this approach's effectiveness and robustness across diverse periodic and chaotic trajectories, demonstrating resilience against noise, disturbances, and system uncertainties.**

Keywords*— Nonlinear tracking control, Model-free control, Machine learning, Reservoir computing, Robotic manipulator, Complex trajectories, Chaotic trajectories, Partial state observation, Stochastic input data.

I. INTRODUCTION

Controlling chaotic or nonlinear systems is a central challenge in engineering, particularly in applications requiring precise tracking of dynamic paths, such as robotic arms tasked with following intricate or chaotic trajectories. Traditional control methods stabilize chaotic trajectories by aligning them with desired periodic paths using small perturbations, typically assuming detailed knowledge of the system's model and equations. However, for high-dimensional, complex dynamics, model-based methods like feedback linearization, backstepping, and sliding mode control face limitations—especially in handling strong nonlinearity, high dimensionality, and delayed responses. As systems grow more intricate, achieving accurate tracking control without full system models becomes increasingly challenging, fuelling interest in model-free, data-driven approaches.

Machine learning (ML) has recently introduced promising avenues for model-free control, allowing adaptive, data-driven systems to track complex trajectories using observational data rather than precise model parameters. By leveraging prior system performance, ML-based control methods generate control laws that improve performance adaptively. While traditional ML control strategies often rely on model-specific assumptions, recent advancements in continuous-time ML have shifted toward more flexible and adaptive control systems suitable for real-time applications.

This study introduces a machine-learning framework utilizing reservoir computing for model-free tracking control of a robotic two-arm manipulator. Designed to operate on partial observational data, the framework tracks complex and chaotic trajectories effectively without relying on explicit system models. By training on consecutive state observations in an offline setting, this approach enables a robust and highly responsive control system. Through this model-free, data-driven method, we aim to extend the capabilities of robotic and control systems in dynamic, unpredictable environments.

II. LITERATURE REVIEW

Research in nonlinear control and chaotic systems has evolved through foundational contributions utilizing both classical and data-driven approaches. Traditional chaotic control focuses on stabilizing chaotic trajectories around unstable periodic orbits with minimal interventions, enabling control without complete system models (Ott et al., 1990; Grebogi & Lai, 1997). Techniques like feedback linearization, Lyapunov redesign, and sliding mode control have been essential for managing complex, high-dimensional systems, though they face limitations with model inaccuracies and time delays (Charlet et al., 1989; Dawson et al., 1994; Furuta, 1990).

Machine learning (ML) introduces adaptive, model-free solutions to address these challenges, particularly in environments where system models are unavailable (Duriez et al., 2017; Ma & Wu, 2020). Reinforcement learning, for example, offers a flexible approach for optimizing control actions adaptively (Recht, 2019). Among ML techniques, reservoir computing stands out for its efficiency and real-time adaptability in nonlinear control, providing a robust framework for tracking chaotic dynamics in high-dimensional systems (Weinan, 2017; Bensoussan et al., 2022). Together, these advancements support innovative applications in the tracking and control of complex, chaotic trajectories.

III. METHODOLOGY

The robotic two-arm system's end effector (located at the end of the second arm) is controlled to follow desired trajectories using a machine learning controller. The full system state is represented by an 8-dimensional vector: $x = [C_X, C_Y, q_1, q_2, \dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2]^T$

where:

- C_X and C_Y : Cartesian coordinates of the end effector,
- q_i : angular position,
- $\dot{q}_i$: angular velocity, and
- $\ddot{q}_i$: angular acceleration of arm i (where i =1, 2).

This setup captures the essential dynamic variables required to control the end effector's precise position in real time.

The observed states are captured in a 4-dimensional vector:

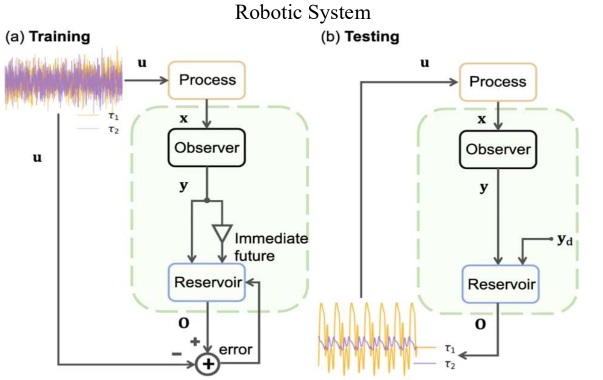
$$y = [C_X, C_Y, \dot{q}_1, \dot{q}_2]^T$$

which is accessible to the machine learning model. The machine learning controller generates a control signal:

$$u(t) = [\tau_1(t), \tau_2(t)]^T$$

representing the torques applied to robotic arms. These torque values are the driving forces that guide the robotic arms along desired paths, enabling the tracking of intricate trajectories.

Figure 1.1 Machine Learning-Based Control Framework for Dual-Arm Robotic System

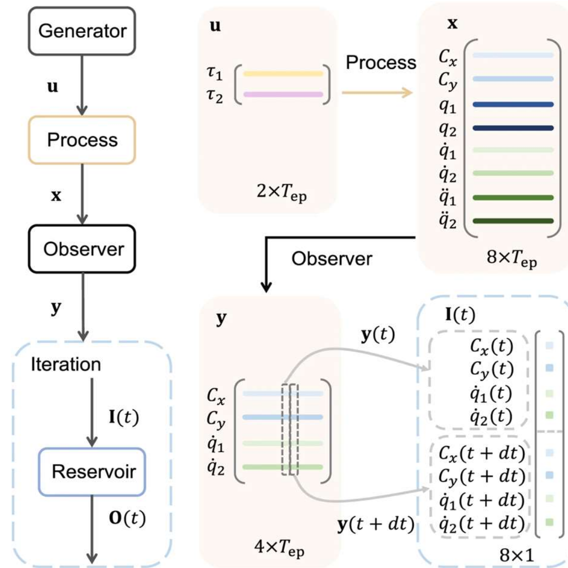


The figure shows training (a) and testing (b) phases of a machine learning control system for a dual-arm robot. In training, random torques $\tau_1(t)$ *and* $\tau_2(t)$ generate state vector x(t), observed as y(t) and fed into a reservoir model. The error between the model output and the original torques trains the system. In testing, the reservoir receives the observed state y(t) and desired state $y_d(t)$, producing control signals to follow the target trajectory.

A. Reservoir Computing Framework

Reservoir computing (RC) is employed as the learning mechanism for the control system. RC is advantageous for dynamic systems due to its efficiency in handling time-sequenced data and reduced training complexity. During training, the RC system receives sequences of observed states $y(t)$ and $y(t + dt)$ as inputs,enabling it to map current observations to the subsequent state using minimal computation. The output of the model is a control signal $O(t)$ designed to align the observed state with the reference trajectory outputs. By leveraging RC, the model can efficiently process time-dependent data, making it well-suited for applications where real-time adaptability is crucial.

Figure 1.2 Data Structure and Variable Flow in Training Phase of Machine-Learning Controller



The figure illustrates the training data structure for a machine-learning controller in a dual-arm robotic system. Random torque signals $\tau_1(t)$ and $\tau_2(t)$ drive the system, creating an $8 \times T_{ep}$ state vector x observed as a $4 \times T_{ep}$ vector y. The reservoir model receives 8-dimensional input combining y(t) and y (t + dt) at each time-step, supporting trajectory learning in the system.

B. State Observation and Training Process

In this phase, the reservoir computing model is trained to recognize dynamic patterns within the system using stochastic control inputs. Each training session is broken down into episodes of length T_{ep}, during which the state variables and $q_1, q_2, \dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2$, as well as the reservoir's internal state, are reset to initial conditions. The control input, $u(t) = [\tau_1(t), \tau_2(t)]^T$, is generated using stochastic signals for a period T_{ep}, resulting in a $2 \times T_{ep}$, torque matrix. Meanwhile, the dynamical state

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

and observed state

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

are collected over the same time frame, resulting in $8 \times T_{ep}$, and $4 \times T_{ep}$,matrices respectively.

The reservoir model receives an 8×1 input vector at every time-step t, comprising the observed state $y(t)$, and its delayed state $y(t + dt)$. The model then learns to generate the control signal that will adjust the system from $y(t)$ to $y(t + dt)$, fulfilling the control objectives over time. This stochastic training ensures extensive coverage of the state space, equipping the model to handle various types of dynamic behaviors. A Gaussian filter smooths the random torque inputs to ensure continuity. The loss function is defined as the difference between the output control actions and the target trajectory, which helps guide the model's learning.

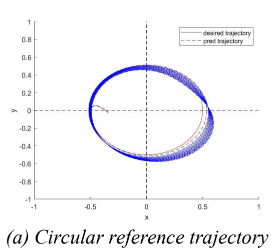
C. Testing and Deployment Phase

In testing, the trained model is tasked with real-time control using a process that inverts the training setup. Given the present state $y(t)$ and the immediate desired future state $y_d(t + dt)$, the reservoir computing model generates a control signal that drives the observed system from $y(t)$ to $y(t + dt)$, minimizing the discrepancy with $y_d(t + dt)$. The model's ability to minimize this discrepancy enables the robotic arm to accurately track diverse trajectories across a range of complexities, including chaotic and nonlinear patterns. The model demonstrates robust adaptability across complex trajectories, including chaotic patterns, while maintaining effective response to disturbances and measurement noise. This adaptability highlights the model's potential for real-time applications in unpredictable or rapidly changing environments.

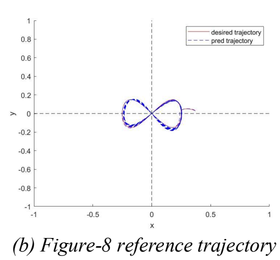
IV. RESULTS

The results demonstrate that the reservoir computing-based control framework effectively enables the robotic system's end effector to track a wide range of complex trajectories with high accuracy. The model was tested on various trajectory types, including both periodic and chaotic patterns, such as circles, figure-eight, Lorenz attractor, and Mackey-Glass chaotic paths. These tests underscore the model's versatility in adapting to different motion types, a key capability for practical deployment in dynamic environments.

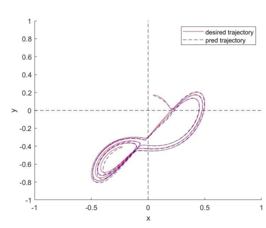
During testing, the reservoir controller utilized the observed current state and target future state to generate control signals that closely matched the desired trajectories, resulting in minimal error. The model exhibited precise adherence to these paths over time, maintaining trajectory alignment even for complex shapes. Consistent alignment with complex trajectories highlights the robustness of the framework in managing both structured and unpredictable patterns, a crucial aspect for real-world control applications. The ability to accurately track both orderly and chaotic patterns showcases the model's flexibility and adaptability in handling diverse dynamic behaviors without requiring explicit system models.



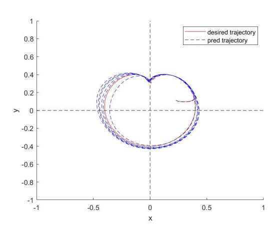
(a) Circular reference trajectory



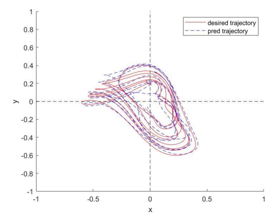
(b) Figure-8 reference trajectory



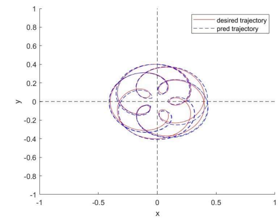
(c) Lorenz reference trajectory



(d) Heart-shaped reference trajectory



(e) Mackey-Glass reference trajectory



(f) Epitrochoid reference trajectory

V. CONCLUSION

In conclusion, this study introduces a machine learning model-free framework utilizing reservoir computing to enable precise trajectory tracking in a robotic two-arm manipulator. By learning control signals from partial state observations, the model effectively tracks a variety of complex trajectories, including chaotic and periodic paths, without relying on a detailed system model. The reservoir-based approach proves adaptable across dynamic behaviors, showcasing strong potential for applications in robotics, where accurate, real-time trajectory tracking is essential. This adaptability, combined with model-free architecture, makes it a viable solution for high-dimensional, nonlinear control challenges across different domains.

Beyond robotics, this framework could be applied to other fields requiring model-free control, such as autonomous vehicles, industrial automation, and soft robotics, where complex, nonlinear dynamics are common. These sectors often face dynamic conditions that would benefit from a control model capable of learning and adapting in real time. The simplicity and adaptability of this model-free approach make it promising for future research to extend machine learning to control high-dimensional, nonlinear systems. The ability to achieve accurate tracking with partial information could inspire further innovations in data-driven control strategies, supporting advancements in autonomous systems, adaptive robotics, and control theory. The framework's success in handling chaotic and dynamic paths suggests it could play a key role in the development of next-generation autonomous systems that demand precision without predefined models.

VI. REFERENCES

[1] Zhai, Zheng-Meng, et al. "Model-free tracking control of complex dynamical trajectories with machine learning." *Nature communications* 14.1 (2023): 5698.

[2] Zhai, Z. -M. Chaotic trajectories. Zenodo https://doi.org/10.5281/ zenodo.8044994 (2023).

[3] Grebogi, C. & Lai, Y.-C. Controlling chaotic dynamical systems. Sys. Cont. Lett. 31, 307–312 (1997).

[4] Grebogi, C. & Lai, Y.-C. Controlling chaos in high dimensions. IEEE Trans. Cir. Sys. 44, 971–975 (1997).

[5] Boccaletti, S., Grebogi, C., Lai, Y.-C., Mancini, H. & Maza, D. Control of chaos: theory and applications. Phys. Rep. 329, 103–197 (2000).

[6] 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)

[7] Klickstein, I., Shirin, A. & Sorrentino, F. Locally optimal control of complex networks. Phys. Rev. Lett. 119,268301(2017).

[8] Jiang, J.-J. & Lai, Y.-C. Irrelevance of linear controllability to nonlinear dynamical networks. Nat. Commun. 10, 3961 (2019).

[9] Aström, K.J. &Murray, R.M. Feedback Systems: An Introduction for Scientists and Engineers 2nd edn (Princeton University Press, NJ, 2021).