Robotica (2025), **43**, pp. 1591–1607
doi:10.1017/S0263574725000578

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Compound Koopman data-driven control for an inchworm robot: validation through virtual experiments

Mehran Rahmani, Jay B. Menon, Apoorva Rahul Uplap and Sangram Redkar

School of Manufacturing Systems and Networks, Ira Fulton School of Engineering, Arizona State University, Mesa, AZ, 85212, USA
Corresponding author: Mehran Rahmani; Email: mrahma61@asu.edu

Received: 25 February 2024; **Revised:** 17 March 2025; **Accepted:** 27 March 2025; **First published online:** 19 May 2025

Keywords: inchworm robot; fractional PID; ISAAC SIM; tracking; compound control

Abstract

Locomotion control of inchworm robots presents significant challenges due to their highly nonlinear dynamics and complex interactions with the environment. Traditional control methods often struggle with achieving precise tracking and adaptive performance in dynamic conditions. To address these limitations, this article proposes a novel data-driven compound control system that integrates fractional proportional-integral derivative (FPID) control with Koopman operator theory. Unlike conventional approaches, which rely on direct nonlinear control or simplified linear approximations, our method leverages data-driven modeling to transform the nonlinear dynamics into a linear representation, making control design more systematic and scalable. A deep neural network is trained to identify the Koopman operator, enabling an FPID controller to operate within this transformed space for improved tracking accuracy and robustness. The proposed framework is validated using NVIDIA Isaac SIM simulation software, demonstrating superior locomotion efficiency and tracking performance compared to existing control strategies. This study advances the control of bio-inspired robots by bridging fractional-order control with data-driven Koopman-based modeling, addressing the fundamental challenge of achieving high-precision locomotion in complex environments.

1. Introduction

Inchworms, also known as geometer caterpillars, exhibit a distinctive mode of locomotion that has inspired robotic motion due to its efficiency and adaptability across various surfaces [1–4]. Their movement is characterized by a rhythmic cycle of contraction and extension, where the inchworm first anchors its hind legs, contracts its body to form an arch, and then extends forward, placing its front legs in a new position. This coordinated movement enables the inchworm to maneuver through complex terrains with precision and minimal energy expenditure [5–7]. By varying the amplitude and frequency of these contractions, inchworms can modulate their speed and adapt to different surface conditions [8], making them ideal natural models for robots that must navigate unpredictable environments. Inspired by this motion, researchers have designed inchworm-like robots that mimic the contraction–extension cycle to achieve efficient, stable locomotion. These robotic systems typically rely on a segmented design with actuators mimicking the inchworm's muscles, allowing controlled body contractions and extensions [9]. By emulating the inchworm's gait, inchworm robots can effectively traverse uneven terrain, squeeze through narrow passages, and adjust their body shape to maintain stability. This biomimetic approach not only improves the robot's adaptability but also enables it to perform in environments where traditional wheeled or legged robots might struggle, such as inside pipes, on soft surfaces, or in confined spaces [10].

Controlling an inchworm robot mechanism requires precise coordination of its segmented body to replicate the natural contraction and extension movement seen in biological inchworms [11]. The robot

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

typically has actuators or motors within each segment, which are programed to alternately contract and extend in a cyclic manner, propelling the robot forward [12]. This type of motion is particularly advantageous for navigating narrow spaces and irregular surfaces, where wheeled or legged robots might face limitations [13]. To ensure accurate positioning and consistent tracking, the control system must dynamically adapt to the robot's changing configuration, continuously adjusting to maintain stability and desired motion patterns [14].

A proportional-integral derivative (FPID) controller can significantly enhance the control of inchworm robot mechanisms by adding flexibility and adaptability to the control process. In a traditional PID controller, the proportional, integral, and derivative terms are adjusted to achieve a balance between response speed and stability, but they are constrained to integer orders [15]. An FPID controller, on the other hand, uses fractional (noninteger) orders for the integral and derivative terms, allowing for finer control tuning. This flexibility enables the FPID controller to handle the nonlinear dynamics and variable interactions between segments of the inchworm robot more effectively [16]. By refining the response to changes in position, force, or external disturbances, the FPID controller helps the inchworm robot maintain smoother, more stable motion, making it highly suited for complex locomotion tasks [17].

Koopman theory offers a powerful framework for analyzing nonlinear dynamical systems by transforming them into a higher-dimensional linear representation, making predicting and controlling complex behaviors easier [18–20]. In Koopman theory, the dynamics of a nonlinear system are represented by a set of observable functions, which capture the system's states and behaviors. Instead of directly tracking the original nonlinear dynamics, the theory utilizes the Koopman operator [21], a linear operator that governs the evolution of these observable functions over time [22]. This transformation allows a nonlinear system to be studied and controlled as if it were linear, enabling the use of linear control techniques for complex systems that are inherently nonlinear, such as robots, fluid dynamics, and autonomous vehicles [23–25]. To generate the Koopman operator in practice, especially for high-dimensional or complex systems, deep neural networks (DNNs) are commonly used. By training a neural network on a set of system data, where each sample contains information on states and their transitions, the DNN learns an approximation of the Koopman operator. In this process, the network is typically designed to map the system's states to a higher-dimensional space where the Koopman operator can act linearly [27]. Once trained, the DNN can predict the evolution of the system's observables, effectively enabling a linear representation of the originally nonlinear dynamics [28]. This combination of Koopman theory and deep learning provides a data-driven approach to modeling, where even complex systems can be controlled with linear methods, enhancing the efficiency and adaptability of the control strategy [29].

A compound control system integrates multiple control methods to leverage the strengths of each approach for improved performance and robustness in complex or dynamic environments [30]. Typically, compound control systems combine a primary control strategy with additional layers, such as adaptive, robust, or learning-based controllers, to handle different system behavior aspects. This layered approach allows the system to respond effectively to both predictable and unpredictable changes, like environmental disturbances or variations in system dynamics [32–36]. The primary benefit of a compound control system is its versatility and resilience in diverse operational conditions. By combining control methods, compound systems can handle a broader range of challenges compared to single-method controllers [37]. They provide better tracking accuracy, improved disturbance rejection, and enhanced adaptability to unknown system variations or external influences. This makes compound control systems particularly valuable in applications requiring high precision and reliability, such as autonomous vehicles, industrial automation, and robotic manipulation [38]. Compound control systems can often simplify tuning, as each component controller addresses specific performance goals, resulting in a more balanced and efficient overall control system.

NVIDIA's Isaac SIM simulation software is a high-fidelity simulation platform designed to support autonomous robot development, testing, and deployment. Built on the NVIDIA Omniverse platform,

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

Isaac SIM provides a physics-based, photorealistic environment where users can simulate robots interacting with complex virtual worlds [39]. It allows for realistic modeling of sensors, actuators, and physical interactions, making it an ideal tool for robotics researchers and developers to prototype, test, and refine their control algorithms and AI-driven behaviors before deploying them on actual hardware. Isaac SIM is particularly useful in scenarios involving perception, navigation, and manipulation, where accurate simulation of real-world conditions is essential. When simulating a robot in Isaac SIM, users can create detailed models of their robots, configure sensor setups, and program control systems to govern the robot's movements and responses. Isaac SIM enables integration with machine learning models and provides a powerful API for managing simulation parameters, enabling developers to run experiments under a wide range of conditions and environments [40]. For instance, a robot's locomotion, obstacle avoidance, and object manipulation tasks can be thoroughly tested in Isaac SIM, with real-time feedback on performance metrics such as tracking accuracy, response to disturbances, and computational efficiency. The software also supports reinforcement learning and data collection for training AI models, allowing for streamlined development of complex autonomous behaviors within a controlled and scalable virtual space [41].

The contributions of this research are as follows:

1. **Development of a novel data-driven compound control system**: This article presents a unique control system designed specifically for the Inchworm robot, which integrates data-driven techniques. Unlike traditional control methods, this system adapts and improves its performance based on real-time data, offering superior precision in handling the robot's movements.
2. **Application of an FPID controller for complex nonlinear dynamics**: A key contribution of this research is the application of a FPID controller to manage the Inchworm robot's complex, nonlinear dynamics. The FPID controller is tailored to effectively address the robot's inherent nonlinearities, which are typically challenging for conventional control strategies.
3. **Introduction of a hybrid control approach using Koopman operator theory**: A novel hybrid control method is proposed, combining traditional control techniques with Koopman operator theory. By transforming the robot's nonlinear dynamics into a data-driven linear model, this approach significantly improves the system's tracking and locomotion performance, overcoming the limitations of purely nonlinear control methods.
4. **DNN-Based Koopman operator identification**: The research introduces an innovative method for identifying the Koopman operator using DNNs. This approach provides an efficient means to approximate the linear dynamics of the Inchworm robot, allowing for better model accuracy and control performance.
5. **Implementation of the FPID controller on the Linearized Koopman model**: After linearizing the robot's dynamics through the Koopman operator, the FPID controller is applied to the transformed model. This enables precise control of the Inchworm robot, leveraging the advantages of both linearization and advanced control theory to enhance its movement accuracy.
6. **Validation through extensive virtual experiments in NVIDIA Isaac SIM simulation software**: The effectiveness of the proposed control system is thoroughly validated through extensive virtual experiments conducted in NVIDIA Isaac SIM. These experiments demonstrate a significant improvement in the robot's locomotion and tracking capabilities, providing strong evidence of the practical applicability of the proposed method.

The remainder of this article is organized as follows: Section 2 discusses the locomotion and dynamic modeling of the Inchworm robot. Section 3 explains the proposed compound control method in detail. Section 4 presents the simulation and virtual experiment results, demonstrating the effectiveness of the control approach. Finally, Section 5 concludes the paper by summarizing the key findings and implications of the study.

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press



Figure 1. *Inchworm in nature [42].*

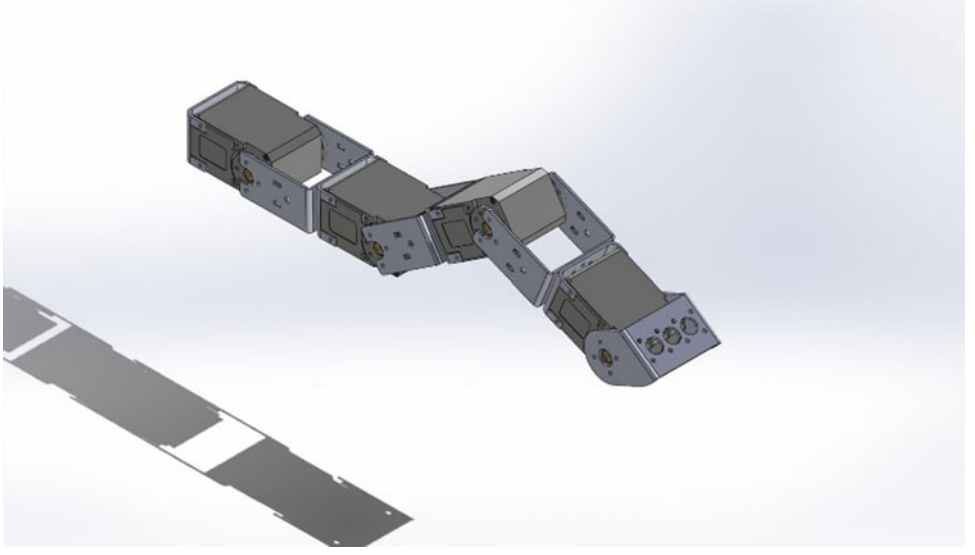


Figure 2. *Inchworm robot mechanism.*

2. Inchworm robot

The Inchworm robot is a bio-inspired robotic system designed to mimic the locomotion of its biological counterpart, the inchworm. This unique robot typically features a segmented body, allowing it to move in a coordinated, wave-like manner (Figure 1). By alternately contracting and extending its segments, the Inchworm robot can traverse various surfaces with high efficiency and adaptability. Its design often incorporates advanced control algorithms, enabling precise movement and the ability to navigate complex environments. The Inchworm robot's applications range from exploration in constrained spaces to tasks in delicate environments, such as search and rescue operations, making it a versatile tool in both research and practical applications. The proposed robot, inspired by the biological mechanics of the inchworm, is illustrated in Figure 2.

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

The four degrees of freedom in an Inchworm robot dynamic model can be defined as [43]:

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{V}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}^2 + \boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{K}\boldsymbol{\tau} + \boldsymbol{\lambda}\boldsymbol{F}_R \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbf{R}^{4\times4}$, $\boldsymbol{V}(\boldsymbol{\theta}) \in \mathbf{R}^{4\times4}$, $\boldsymbol{G}(\boldsymbol{\theta}) \in \mathbf{R}^{4\times1}$, and $\boldsymbol{K} \in \mathbf{R}^{4\times1}$ are mass matrix, centrifugal coefficient matrix, gravity vector, and subtraction matrix. Also, the $\boldsymbol{F}_{\mathbf{R}}$ is forces that apply from the ground. The comprehensive dynamic model, including all relevant equations and parameters, is thoroughly detailed in reference [30]:

$$\boldsymbol{M}(\boldsymbol{\theta}) = \frac{\boldsymbol{ml}^2}{\mathbf{6}}\begin{bmatrix} \mathbf{20} & \mathbf{15}\boldsymbol{c}_{12} & \mathbf{9}\boldsymbol{c}_{13} & \mathbf{3}\boldsymbol{c}_{14} \\ \mathbf{15}\boldsymbol{c}_{12} & \mathbf{14} & \mathbf{9}\boldsymbol{c}_{23} & \mathbf{3}\boldsymbol{c}_{24} \\ \mathbf{9}\boldsymbol{c}_{13} & \mathbf{9}\boldsymbol{c}_{23} & \mathbf{8} & \mathbf{3}\boldsymbol{c}_{34} \\ \mathbf{3}\boldsymbol{c}_{14} & \mathbf{3}\boldsymbol{c}_{24} & \mathbf{3}\boldsymbol{c}_{34} & \mathbf{2} \end{bmatrix} \tag{2}$$

$$\boldsymbol{V}(\boldsymbol{\theta}) = \frac{\boldsymbol{ml}^2}{\mathbf{6}}\begin{bmatrix} \mathbf{0} & \mathbf{15}\boldsymbol{s}_{12} & \mathbf{9}\boldsymbol{s}_{13} & \mathbf{3}\boldsymbol{s}_{14} \\ -\mathbf{15}\boldsymbol{s}_{12} & \mathbf{0} & \mathbf{9}\boldsymbol{s}_{23} & \mathbf{3}\boldsymbol{s}_{24} \\ -\mathbf{9}\boldsymbol{s}_{13} & -\mathbf{9}\boldsymbol{s}_{23} & \mathbf{0} & \mathbf{3}\boldsymbol{c}_{34} \\ -\mathbf{3}\boldsymbol{s}_{14} & -\mathbf{3}\boldsymbol{s}_{24} & -\mathbf{3}\boldsymbol{s}_{34} & \mathbf{0} \end{bmatrix} \tag{3}$$

$$\boldsymbol{G}(\boldsymbol{\theta}) = \frac{\boldsymbol{mgl}}{\mathbf{2}}\begin{bmatrix} \mathbf{7}\,\boldsymbol{cos}\,\boldsymbol{\theta}_1 \\ \mathbf{5}\,\boldsymbol{cos}\,\boldsymbol{\theta}_2 \\ \mathbf{3}\,\boldsymbol{cos}\,\boldsymbol{\theta}_3 \\ \boldsymbol{cos}\boldsymbol{\theta}_4 \end{bmatrix} \tag{4}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} \mathbf{1} & -\mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & -\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & -\mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{5}$$

In the described system, θ_i represents the angle of the ith joint. The expressions $c_{ij} = \cos(\theta_i - \theta_j)$ and $s_{ij} = \sin(\theta_i - \theta_j)$ denote the cosine and sine of the angle difference between the ith and jth joints, respectively. Furthermore, m denotes the mass of each link, while l specifies the length of each link in the system.

3. Proposed compound control

A compound control system combining FPID control, Koopman theory, and DNNs presents a sophisticated approach for managing the nonlinear dynamics of an Inchworm robot, whose bio-inspired design and movement require precise and adaptable control. FPID control, with its ability to incorporate fractional-order derivatives, enhances tuning flexibility and robustness against disturbances, making it well-suited for the Inchworm robot's variable-speed, segmented movement. By integrating Koopman theory, this system transforms the robot's nonlinear motion dynamics into a linear framework, simplifying complex dynamics and enabling the application of linear control techniques for smoother, more predictable operation. DNNs complement this by learning and approximating unknown nonlinear functions within the system, providing real-time adaptability and refinement to the control strategy as the robot encounters diverse environments and terrains. This compound approach equips the Inchworm robot with a resilient and efficient control system that combines precision, adaptability, and robustness, crucial for achieving effective and stable locomotion in dynamic applications.

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

3.1 Fractional PID control

Fractional proportional-integral derivative control offers a robust framework for managing the nonlinear dynamics of an inchworm robot. Unlike traditional PID controllers, FPID utilizes fractional calculus, allowing more flexible and precise adjustments in response to the robot's complex motion patterns. This control strategy can handle the inherent time delays, varying friction, and highly nonlinear interactions among the joints and links of the Inchworm robot, leading to improved trajectory tracking and disturbance rejection. By tuning the fractional orders of the derivative and integral terms, FPID control achieves a balance between responsiveness and stability, making it particularly effective for tasks requiring precise positioning and adaptability. For the Inchworm robot, FPID enhances control over joint movements and minimizes tracking errors, contributing to smoother, more efficient locomotion in real-world applications. The FPID control can be defined as

$$u_{FOPID} = K_p e(t) + K_i D^{-\mu} e(t) + K_d D^{\mu} e(t) \tag{6}$$

where $e(t) = \theta_d - \theta$, which θ_d is desired trajectory, D is fractional operator defines as $D = \frac{d}{dt}$, and μ is fractional order. The Grunwald–Letnikov [44] approach to fractional calculus defines a fractional derivative or integral based on discrete, incremental changes, making it suitable for practical computations and numerical applications. It generalizes the concept of integer-order differentiation and integration to non-integer orders, allowing more flexibility in modeling complex, memory-dependent systems that classical calculus cannot capture. The Grunwald–Letnikov fractional derivative of order α at a point t is derived as a limit of a summation involving differences of function values, weighted by binomial coefficients, which reflect the memory effect typical of fractional systems.

$$D_t^{\mu} e(t) = \lim_{h \to 0} h^{-\mu} \sum_{k=0}^{n} (-1)^k \binom{\mu}{k} f(e(t) - kh) \tag{7}$$

where

$$\binom{\mu}{k} = \frac{\mu(\mu - 1)(\mu - 2) \cdots (\mu - k + 1)}{k!} = \frac{\Gamma(\mu + 1)}{k!\Gamma(\mu - k + 1)} \tag{8}$$

Detailed explanations can be found in [44]. The primary design parameters of a FPID Controller –K_p, K_i, K_d, λ, and μ significantly influence control performance. K_p determines response speed and sensitivity while K_i eliminates steady-state error but may introduce instability if too high. K_d improves damping and reducing overshoot but increases noise sensitivity. The fractional orders λ and μ (derivative) provide added flexibility, enhancing disturbance rejection and response smoothness. Higher λ strengthens long-term correction, whereas higher μ improves damping but may cause excessive smoothing. Proper tuning balances speed, stability, and robustness, making FPID superior to classical PID in adaptability and performance.

3.2 Koopman theory

Koopman theory is a mathematical framework used to analyze nonlinear dynamical systems by transforming them into a higher-dimensional linear framework, which can simplify control and analysis. The Koopman operator, K, is an infinite-dimensional linear operator that acts on observable functions of the state of a nonlinear dynamical system. For a dynamical system described by [27]:

$$x_{t+1} = f(x_t) \tag{9}$$

where x_t is the state vector at time t and f is a (nonlinear) function defining the system's dynamics. The Koopman operator acts on observables rather than on states. An observable, g, is a function that maps the state x to some real or complex value, $g(x)$. Then, the Koopman operator K is defined by

$$Kg(x) = g(f(x)) \tag{10}$$

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

This means the Koopman operator propagates observables forward in time. In Koopman theory, the goal is to lift the system into a higher-dimensional space using observables where the dynamics can be approximated as linear. For a vector of observables

$$g(x) = \left[g_1(x), g_2(x), \ldots, g_n(x)\right]^T \tag{11}$$

The evolution can be represented as

$$g(x_{t+1}) = kg(x_t) \tag{12}$$

If the Koopman operator K is well-approximated by a finite-dimensional matrix K, this matrix can describe the evolution of the observables in a linear fashion:

$$g(x_{t+1}) \approx Kg(x_t) \tag{13}$$

Koopman theory leverages the eigenfunctions of the Koopman operator to analyze the system's dynamics. An eigenfunction $\phi(x)$ of the Koopman operator satisfies

$$K\phi(x) = \lambda\phi(x) \tag{14}$$

where λ is the eigenvalue associated with the eigenfunction ϕ. These eigenfunctions are particularly valuable because they provide insights into the invariant structures of the system, and the eigenvalues can provide information about the system's stability and time scales.

3.3 Deep neural network

In using a DNN to generate the Koopman operator, the key is to map a nonlinear system into a higher-dimensional space where linear dynamics can approximate the system's behavior. A DNN can achieve this by learning complex mappings that approximate Koopman eigenfunctions, which represent the system's state in a transformed "Koopman space." In this space, the dynamics can be represented as a linear, time-invariant system governed by the Koopman operator. This linearization allows even nonlinear systems to be analyzed, controlled, or predicted with linear techniques, greatly simplifying the control design. To accomplish this, a typical DNN structure involves an encoder-decoder setup. The encoder maps the original system states into the latent Koopman space, capturing the necessary eigenfunction approximations, while the decoder reconstructs the original states from this transformed space. Training the DNN to both approximate and reconstruct the system states ensures that essential system dynamics are preserved in the Koopman space, where the Koopman operator linearly propagates future states. This process enables predictions and control that would otherwise be difficult to achieve with highly nonlinear dynamics.

The Koopman Neural Network leverages a convolutional neural network (CNN) to approximate the Koopman operator, integrating deep learning's capabilities to estimate the Koopman matrix across dynamic systems robustly. By refining model parameters iteratively, this approach optimizes the computation of the Koopman operator at each time step, enabling the network to adapt and learn more precise representations of the system dynamics. CNNs are particularly advantageous here due to their inherent ability to detect and capture spatial dependencies and patterns within high-dimensional data, which is crucial for recognizing intricate dynamics present in complex systems. Through its layered structure, the CNN component progressively extracts features from the input signals, learning latent representations that reflect the system's behavior over time. This process equips the network to capture immediate spatial relationships and build a comprehensive understanding of temporal dependencies, enhancing the accuracy of Koopman operator estimates. As the network trains on representative datasets, it continuously refines its internal parameters, leading to more accurate, generalizable predictions that closely mirror the system's real-world dynamics. Moreover, the iterative optimization in the Koopman neural network model fosters predictive accuracy while also allowing for adaptive updates based on real-time feedback. This adaptability is vital for real-time control applications and systems requiring responsive adjustments, as it enables the model to dynamically account for shifts or disturbances in system behavior. Consequently, this approach advances the capability to perform adaptive control, where the system

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

can continuously fine-tune its behavior in response to new data, making the Koopman neural network a powerful tool for capturing and leveraging the complexities of dynamic systems in high-dimensional settings.

In dynamical systems modeling, a system's state is typically recorded as a sequence of snapshots taken at consecutive time steps. We denote these states in a matrix X, where each column X_i represents the system's state at time t_i. Thus, we construct $X = [X_i, \ldots, X_{n-1}]$, with n representing the total time steps considered. To train a neural network effectively, input–output pairs are essential. Here, we use the current state X_i as the input and the subsequent state X_{i+1} as the corresponding output. This setup leverages the worm's metachronal gait, ensuring that the position of actuator $j+1$ at time $t+1$ corresponds to the position of actuator j at time t. Consequently, the training dataset is structured as $X_{train} = [X_i, \ldots, X_{n-1}]$ and $Y_{train} = [X_2, \ldots, X_n]$.

The Koopman operator, symbolized by K, offers a powerful approach for analyzing the time evolution of complex dynamical systems in discrete time steps by leveraging the concept of observables. Observables are functions that map the original, often nonlinear, state of the system into a higher-dimensional space, where the dynamics can be represented in a linear form. In this framework, the system's evolution is expressed by the relation $f(X_{t+1}) = K.f(X_t)$, where $f(.)$ denotes an observable that captures key features of the state X_t at time t. Applying the Koopman operator to these observables makes it possible to achieve a linear representation of otherwise nonlinear dynamics, enabling more straightforward analysis and control strategies. The key insight behind this approach is the concept of "lifting" – embedding the system's state space into a higher-dimensional observable space where the system's evolution can be represented through linear operators. This lifting process effectively transforms complex nonlinear interactions within the system into a more accessible linear form, even though nonlinear rules may govern the original system. By bridging the gap between nonlinear state-space dynamics and a linear observable space, the Koopman operator framework enhances our ability to model, analyze, and control intricate dynamical systems. It is a valuable tool in fields ranging from robotics to fluid dynamics. We typically rely on a comprehensive dataset capturing the system's state evolution over time to compute the Koopman operator. Let's assume we have a training dataset, where X_{train} represents the current state of the system across various time steps, and Y_{pred} represents the future states corresponding to each entry in X_{train}. In this context, the relationship between these current and future states can be formulated as $Y_{pred} = K.X_{train}$, where K denotes the Koopman operator. We aim to calculate K such that it optimally maps X_{train} to Y_{pred}, capturing the system's underlying dynamics in a linear transformation that applies to potentially nonlinear systems. By accurately estimating K, we enable predictions of future states based solely on current observations, which is foundational in extending Koopman theory to control and optimization tasks.

This calculation leverages the Moore–Penrose pseudoinverse to approximate the Koopman operator, an essential component in data-driven dynamic modeling. Let X_{train} be a matrix representing the system's states over time, capturing the system's dynamics in various sampled configurations. To obtain an approximate solution that minimizes errors in a least-squares sense, we first compute the pseudoinverse of X_{train}, denoted as X^{+}_{train}. This pseudoinverse satisfies the relationship $X_{train} = I$, where I is the identity matrix, ensuring that the inverse-like properties hold even for nonsquare matrices. With X^{+}_{train} computed, we then calculate the Koopman operator K by applying the expression $K = Y_{pred}.X^{+}_{train}$, where Y_{pred} represents the future state predictions of the system. The result is an approximation of the Koopman operator that provides the best possible mapping from the current state to the future state under the least-squares criterion, allowing the model to capture the essential dynamics of the system effectively. This method facilitates a robust and efficient data-driven approach to estimate the linear dynamics of even complex, nonlinear systems.

Using the pseudoinverse in this method provides robustness, particularly in cases where X_{train} is not square or is rank-deficient. This approach enables reliable approximation of the Koopman operator even under challenging data conditions, making it highly valuable for practical applications across various domains. The Koopman operator's capacity to capture and linearize complex, nonlinear dynamics has transformed analysis, prediction, and control in fields as diverse as fluid dynamics, robotics,

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

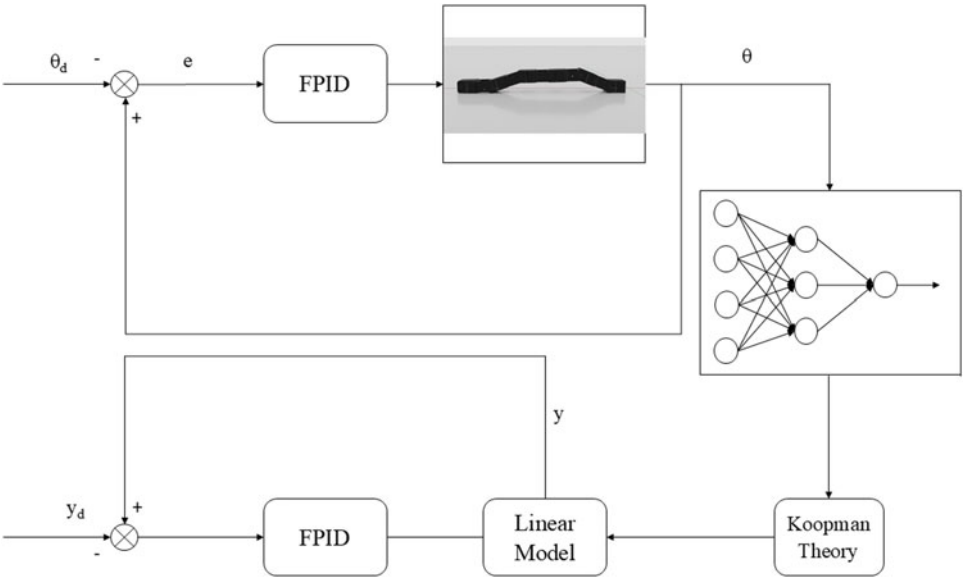


Figure 3. *The proposed control method block diagram.*

and beyond. By transforming nonlinear systems into an approximate linear framework, the Koopman operator enhances the computational efficiency and accuracy of predictive models and controllers for dynamic systems. Once the Koopman operator is applied, the resulting linear dynamical system can be represented as follows, enabling powerful insights into system behavior that would otherwise require computationally intensive nonlinear approaches. This formulation allows for the design of linear controllers, which are often easier to implement and tune, providing a streamlined solution for managing complex, nonlinear processes.

$$\frac{dy}{dt} = Ky + u \tag{15}$$

The proposed control block diagram is shown in Figure 3. The input for the DNNs is derived from solving the complex nonlinear dynamics model of an Inchworm robot. By analyzing the system's behavior through this nonlinear model, comprehensive input data for the DNNs is generated, capturing intricate dynamic patterns (size of data is 4×1 matrix). Following this data generation process, an FPID controller is applied to the worm robot's resulting data-driven linear dynamic model. This approach enables the FPID controller to leverage the data's underlying nonlinear dynamics, enhancing the control precision, stability, and overall performance. The integration of DNNs with the FPID controller provides a robust control strategy, effectively improving the robot's tracking accuracy and response under various conditions. The research employs several key strategies to manage uncertainties in system parameters, including using data-driven control methods that adapt to real-world conditions, implementing an FPID controller to handle nonlinear dynamics and improve robustness, and a hybrid control approach utilizing Koopman operator theory to linearize the system. Additionally, a DNN is trained to identify the Koopman operator, enhancing the accuracy of the linearized model. The FPID controller is then applied to the transformed model, reducing complexity and increasing predictability. Finally, extensive virtual experiments in NVIDIA Isaac SIM Software validate the method's effectiveness under various conditions, ensuring robust performance despite uncertainties.

The stability of the fractional PID-controlled system is analyzed using a Lyapunov function $V(e) = \frac{1}{2}e^2$. Taking its fractional derivative and substituting the system's dynamics, we obtain $D^{\alpha}V(e) = -K_p e^2 - K_i e D^{-\mu} e - K_d e D^{\mu} e$. Since all terms are negative for $K_p, K_i, K_d > 0$, the Lyapunov function is nonincreasing, ensuring the system's bounded error and asymptotic stability. This proves that the fractional PID controller effectively stabilizes the Inchworm robot's motion. The proposed control system

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

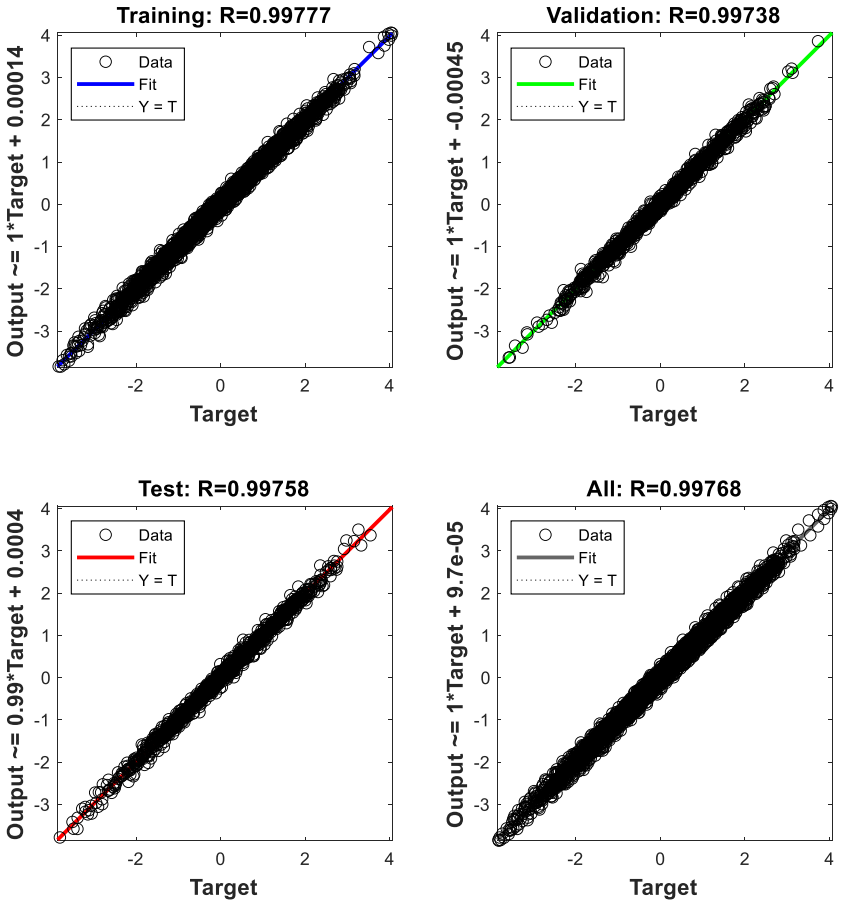


Figure 4. *Regression analysis for proposed neural network.*

may face instability under high-frequency disturbances, rapidly changing external forces and extreme nonlinearities that the Koopman-based model fails to capture. Its performance heavily depends on the quality of training data, making it less robust to untrained disturbances. The FPID controller's sensitivity to parameter tuning and the computational overhead of the hybrid approach could also introduce delays, impacting real-time stability. Additionally, since validation is conducted in simulation, real-world uncertainties such as sensor noise and actuator imperfections remain untested, posing potential challenges in practical deployment.

4. Virtual experiment results

This study introduces a novel data-driven control approach for managing an Inchworm robot. The simulation's initial conditions are outlined as follows:

$$\theta_0 = \dot{\theta}_0 = [0, 0, 0, 0] \qquad y_0 = \dot{y}_0 = [0, 0, 0, 0]$$

The desired trajectory tracking is as follows:

$$\theta_{1d} = y_{1d} = 0.2 \sin(t)$$
$$\theta_{2d} = y_{2d} = 0.4 \sin(2t)$$
$$\theta_{3d} = y_{3d} = 0.6 \sin(3t)$$
$$\theta_{4d} = y_{4d} = 0.8 \sin(4t)$$

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

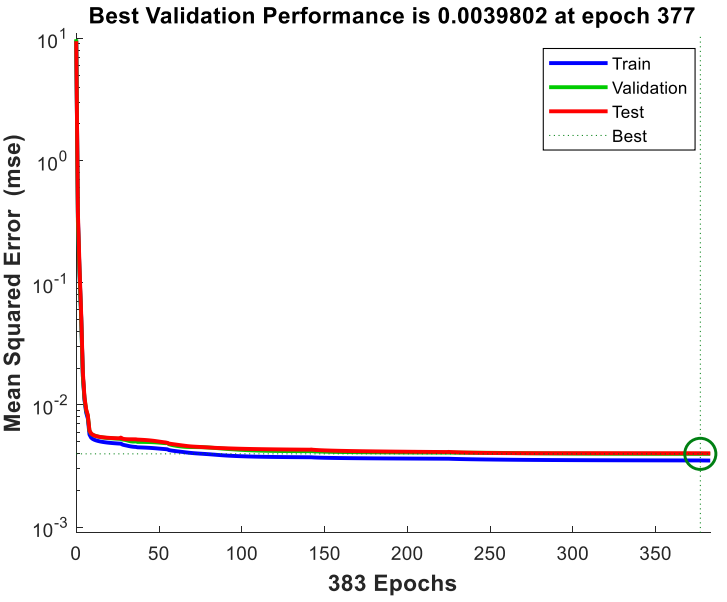


Figure 5. *Mean squared error analysis for the proposed neural network.*

The fractional order parameters are as follows:

$$\alpha = 0.01, \quad \mu = 0.75, \quad \text{and} \quad h = 0.01.$$

Koopman operator in Figure 4 presents a detailed regression plot analysis that illustrates the performance of the neural network model in predicting the Koopman eigenfunctions. This plot reveals a remarkably robust alignment between the eigenfunction values predicted by the neural network and the actual eigenfunction values derived from the dataset. Notably, the data points in the plot demonstrate a tight clustering around the diagonal line, which is a strong indicator of the neural network's proficiency in accurately capturing the underlying dynamics of the system. This close adherence to the diagonal signifies that the predictions made by the neural network are close to the actual values and reflect the system's behavior with a high degree of fidelity. Moreover, the strong linear trend observed in the plot, combined with minimal scatter among the data points, further underscores the efficacy of the neural network in estimating the Koopman operator. This indicates that the model is effectively uncovering the latent dynamics that govern the complex behavior of the system. Thus, Figure 4 serves as compelling evidence of the neural network's success in estimating the Koopman operator, showcasing its potential to advance our understanding and analysis of intricate dynamical systems significantly. The implications of this success extend beyond mere predictive capability, suggesting that such neural network approaches could facilitate deeper insights into the behaviors of various systems, ultimately contributing to the fields of dynamical systems theory and data-driven modeling.

In Figure 5, the mean square error (MSE) plot plays a pivotal role in evaluating the neural network's performance tasked with estimating the Koopman operator. This plot meticulously charts the progression of MSE values throughout the training process, offering valuable insights into the model's ability to minimize the error between its predictions and the actual Koopman eigenfunctions. A key aspect of the MSE plot is its trend over time. A consistent decrease in MSE values indicates that the neural network is effectively refining its predictions. This downward trajectory suggests that the model is learning to approximate better the Koopman operator, which is fundamental for understanding the dynamics of the system being analyzed. As the MSE diminishes, it reflects an improvement in the model's performance, demonstrating its increasing capability to capture the underlying dynamics with greater accuracy. Furthermore, a low and stable MSE value during the latter stages of training is particularly significant.

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

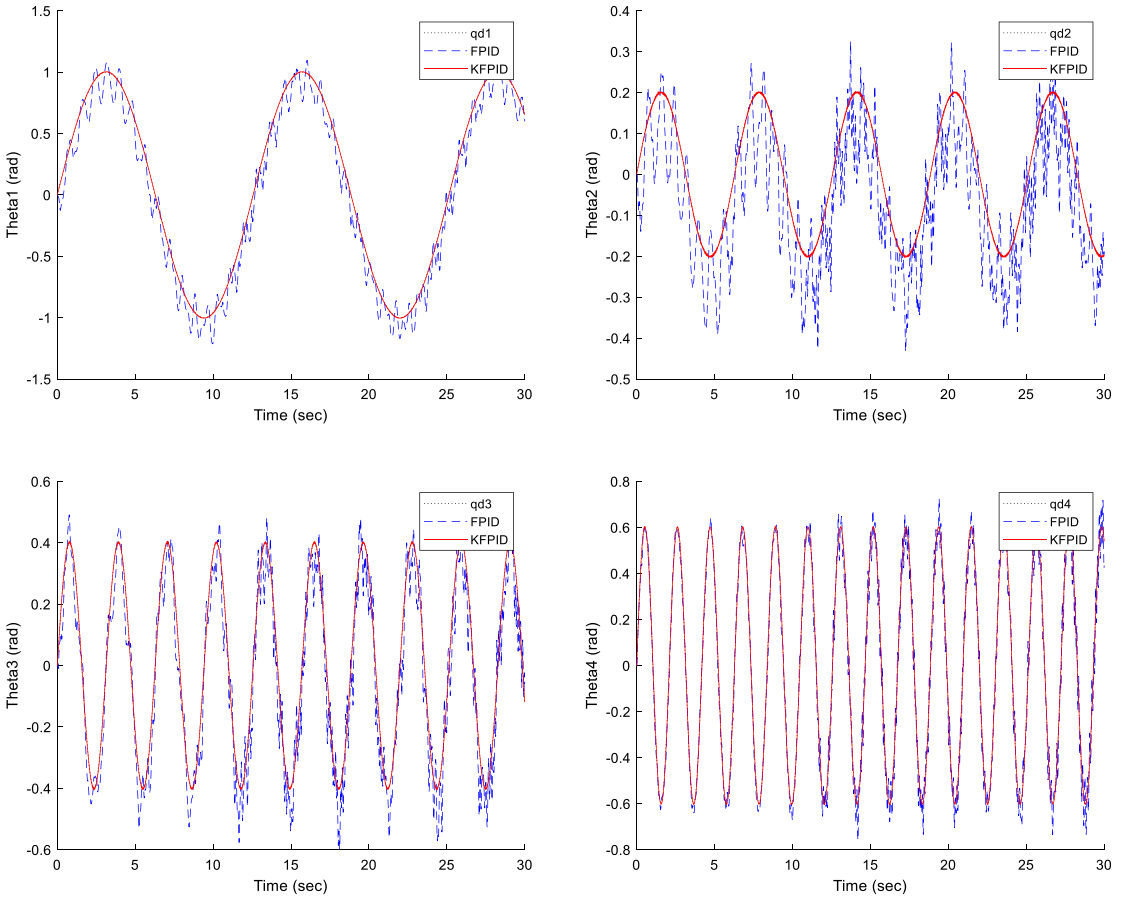


Figure 6. *The position tracking of the inchworm robot joints under the proposed controller.*

It implies that the model has likely converged towards a satisfactory solution, suggesting that it can reliably estimate the Koopman operator. This convergence is critical, as it indicates that the neural network has achieved a level of precision that allows it to represent the system's dynamics effectively. In essence, Figure 5 serves as compelling evidence of the neural network's proficiency in estimating the Koopman operator. The behavior of the MSE plot throughout the training process not only illustrates the learning journey of the model but also underscores its potential utility in dynamic systems analysis. Thus, the insights gleaned from the MSE plot are instrumental in validating the effectiveness of the neural network in this context.

Figure 6 illustrates the tracking performance of the Inchworm robot joints when utilizing the proposed control method. The results demonstrate that the KFPID controller significantly outperforms the traditional FPID controller in terms of tracking accuracy. The KFPID controller leverages a novel approach by first generating data from the nonlinear dynamic model of the Inchworm robot during its initial control loop. These data are crucial for training DNNs to identify the Koopman operator, which provides a linear representation of the robot's dynamics. By capturing the essential features of the nonlinear system, the Koopman operator allows us to derive a linear data-driven dynamic model. Once this linear model is established, the FPID controller can be applied to it. This transition from a nonlinear to a linear model enhances the controller's performance because linear systems are generally easier to control and analyze. The improved tracking performance observed with the KFPID controller can be attributed to its ability to effectively utilize the linear approximation of the system, thereby reducing the tracking error and enhancing responsiveness. In summary, the integration of the Koopman operator into the control

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

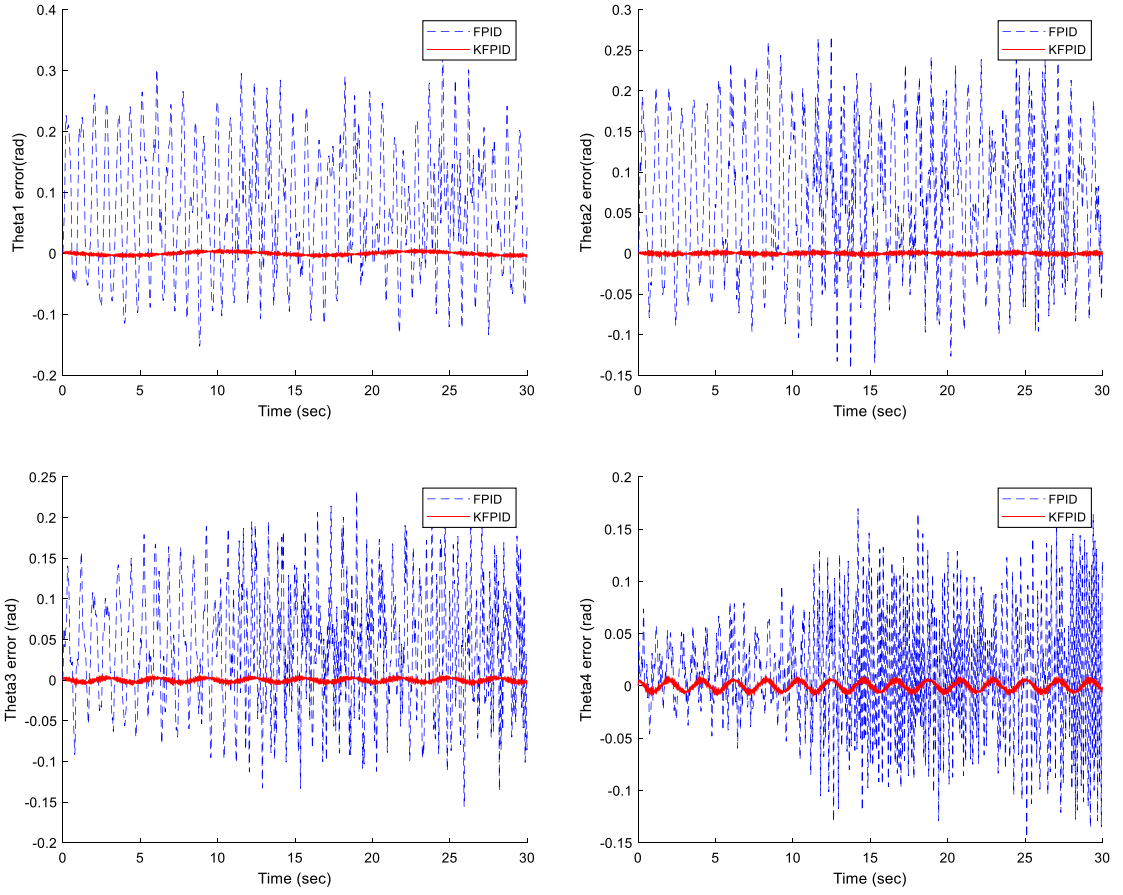


Figure 7. *The position tracking error of the worm robot joints under the proposed controller.*

strategy not only facilitates the extraction of a linear dynamic model but also enables the application of more sophisticated control techniques. This results in superior tracking performance for the Inchworm robot, highlighting the advantages of combining traditional control methods with modern data-driven approaches.

Figure 7 presents the tracking error of the Inchworm robot joints when using the KFPID controller. The results indicate that the KFPID controller achieves significantly lower tracking errors compared to the traditional FPID controller. The reduced tracking error can be attributed to the KFPID controller's ability to model the complex dynamics of the Inchworm robot more effectively. By incorporating the Koopman operator, the KFPID controller benefits from a data-driven linear representation of the system, which allows for improved control accuracy and responsiveness. This approach enables the controller to adapt more effectively to system behavior variations and external disturbances. In contrast, the FPID controller, which operates on nonlinear dynamics without the benefit of the Koopman operator, may struggle to capture the nuances of the robot's movements, leading to higher tracking errors. The KFPID controller's fractional-order component further enhances its ability to fine-tune the response characteristics, providing a more nuanced control that better aligns with the desired trajectory. Overall, the data presented in Figure 7 underscore the effectiveness of the proposed KFPID controller in minimizing tracking errors, thereby improving the overall performance of the Inchworm robot.

Figure 8 illustrates the control input signals generated by the proposed KFPID controller in comparison to an FPID controller. Notably, the KFPID controller exhibits significantly less control effort, which is a critical metric for evaluating controller efficiency. Control effort refers to the amount of input signal

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

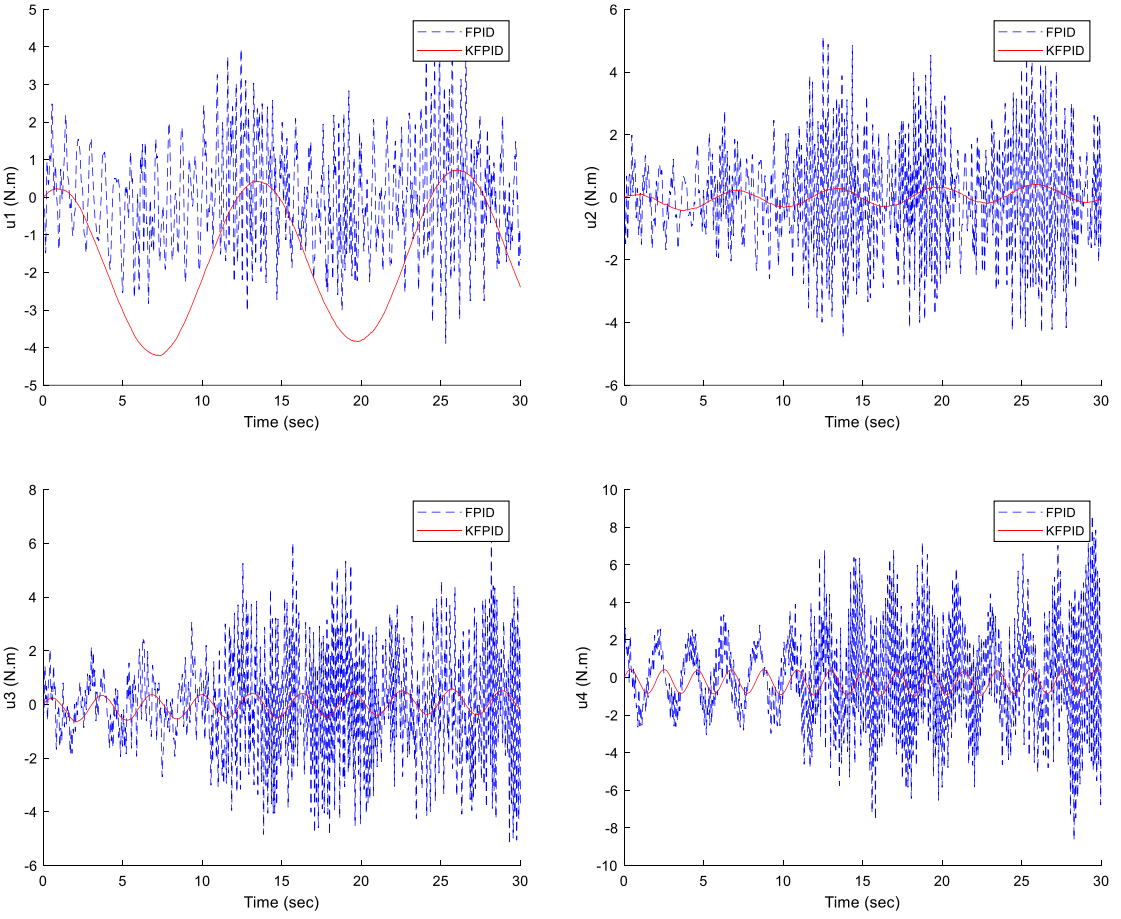


Figure 8. *The control input signals of the inchworm robot joint under the proposed controller.*

exerted by the controller to achieve desired system performance; lower values indicate a more efficient strategy. The smoother control input signals from the Koopman controller suggest a more effective operation with fewer oscillations. In contrast, the conventional FPID controller displays greater fluctuations, indicative of a more aggressive approach that can lead to increased wear on actuators and reduced energy efficiency. The incorporation of the Koopman operator allows for capturing the system's dynamics in a higher-dimensional space, enabling better predictions and optimized control actions, ultimately reducing the need for excessive adjustments. This reduction in control effort enhances system stability and lowers the risk of overshoot and instability during disturbances or setpoint changes.

The virtual experiment depicted in Figure 9 demonstrates the locomotion of the Inchworm robot utilizing the proposed control method, executed within the Isaac Sim software environment. This simulation effectively showcases the robot's movement dynamics, allowing for a detailed analysis of its inching motion. To begin with, the inchworm robot's design was accurately represented through a CAD file, which was imported into Isaac Sim. This step ensured that the robot's physical characteristics and kinematic properties were faithfully modeled, providing a solid foundation for the simulation. The simulation highlights the robot's unique locomotion mechanism, where the coordinated contraction and extension of its segments mimic the natural movement of an inchworm. The control method implemented in this experiment allows for precise modulation of the robot's joints and segments, enabling smooth transitions and effective ground engagement. Analyzing the results reveals several key insights. The simulation captures the incremental movements of the robot, demonstrating how each segment effectively grips and releases the surface, facilitating forward propulsion. The interaction between the

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

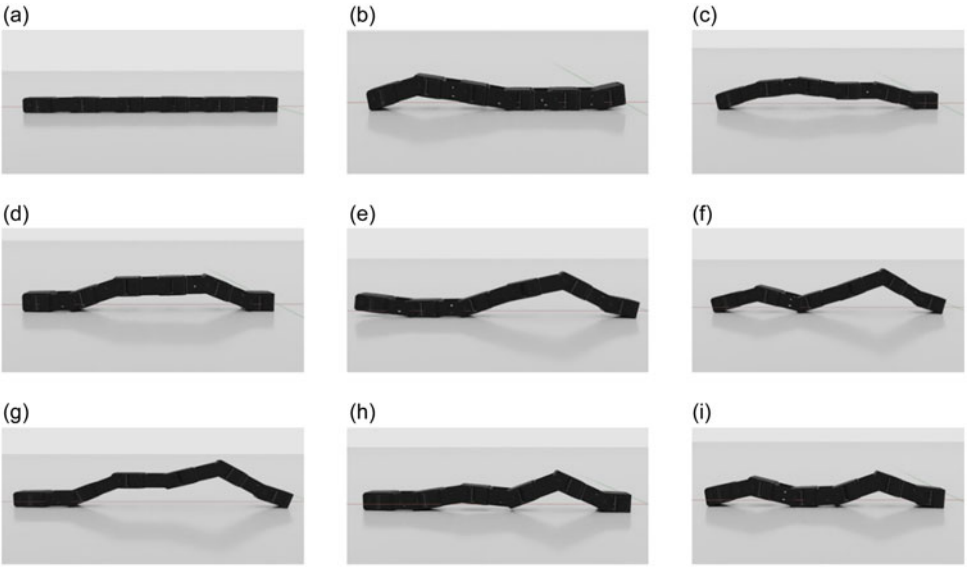


Figure 9. *The virtual experiment of the inchworm robot in Isaac SIM software.*

robot's body and the environment is critical, as it influences the efficiency and speed of locomotion. The proposed control method is validated through its ability to adapt the robot's movements in real time. The software provides feedback on the robot's performance, indicating how well the control strategy responds to different terrains and obstacles. Various performance metrics, such as speed, stability, and energy consumption, can be analyzed through the simulation data. This quantitative analysis helps assess the control method's effectiveness and potential applications in real-world scenarios. The use of Isaac Sim allows for a rich visual representation of the robot's motion, which can be instrumental for further development and refinement of the control algorithms. The graphical feedback can guide modifications to improve locomotion efficiency. Overall, the virtual experiment serves as a compelling demonstration of the Inchworm robot's capabilities under the proposed control method, providing valuable insights for further research and development in robotic locomotion strategies.

5. Conclusion

This article introduces a novel hybrid control approach for the Inchworm robot, combining an FPID controller with a data-driven Koopman operator model. The method significantly enhances tracking performance and locomotion stability by transforming the robot's nonlinear dynamics into a linear framework. Validation through extensive virtual experiments in NVIDIA Isaac SIM simulation software demonstrates the robustness and effectiveness of the approach. The proposed framework improves control accuracy and adaptability, and its successful validation in simulations confirms its potential for real-world applications. Future research will focus on refining the model, conducting experiments on a physical robot, and exploring its use for other robotic systems.

Author contribution. All authors contributed equally to this article.

Financial support. No financial support.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

References

[1] Y. Liu, Y. Qin, D. Yuan and W. Yuan, "Friction modulation through normal vibrations in an inchworm-inspired robot," *Tribology Int.*, 109837 (2024).

[2] Y. Tian, Y. A. Yao, W. Ding and Z. Xun, "Design and locomotion analysis of a novel deformable mobile robot with worm-like, self-crossing and rolling motion," *Robotica* **34**(9), 1961–1978 (2016).

[3] L. Douadi, D. Spinello, W. Gueaieb and H. Sarfraz, "Planar kinematics analysis of a snake-like robot," *Robotica* **32**(5), 659–675 (2014).

[4] Y. Peng, H. Nabae, Y. Funabora and K. Suzumori, "Controlling a peristaltic robot inspired by inchworms," *Biomimetic Intell. Rob.* **4**(1), 100146 (2024).

[5] M. Li, G. Wang, J. Wang, Y. Zheng and X. Jiao, "Development of an inchworm-like soft pipe robot for detection," *Int. J. Mech. Sci.* **253**, 108392 (2023).

[6] J. Sun, L. Bauman, L. Yu and B. Zhao, "Gecko-and-inchworm-inspired untethered soft robot for climbing on walls and ceilings," *Cell Reports Phys. Sci.* **4**(2), 101241 (2023).

[7] T. H. Lin, A. Putranto, P. H. Chen, Y. Z. Teng and L. Chen, "High-mobility inchworm climbing robot for steel bridge inspection," *Autom. Constr.* **152**, 104905 (2023).

[8] S. Thomas, P. Germano, T. Martinez and Y. Perriard, "An untethered mechanically-intelligent inchworm robot powered by a shape memory alloy oscillator," *Sens. Actuators A: Phys.* **332**, 113115 (2021).

[9] S. Ueno, K. Takemura, S. Yokota and K. Edamura, "Micro inchworm robot using electro-conjugate fluid," *Sens. Actuators A: Phys.* **216**, 36–42 (2014).

[10] M. Rahmani, A. Ghanbari and M. M. Ettefagh, "Hybrid neural network fraction integral terminal sliding mode control of an inchworm robot manipulator," *Mech. Syst. Signal Process.* **80**, 117–136 (2016).

[11] J. Cao, W. Liang, Y. Wang, H. P. Lee, J. Zhu and Q. Ren, "Control of a soft inchworm robot with environment adaptation," *IEEE Trans. Ind. Electron.* **67**(5), 3809–3818 (2019).

[12] Z. Zheng, P. Kumar, Y. Chen, H. Cheng, S. Wagner, M. Chen and J. C. Sturm, "Piezoelectric soft robot inchworm motion by tuning ground friction through robot shape: Quasi-static modeling and experimental validation," *IEEE Trans. Rob.* **40**, 2339–2356 (2024).

[13] Y. Yang, D. Li and Y. Shen, "Inchworm-inspired soft robot with light-actuated locomotion," *IEEE Rob. Autom. Lett.* **4**(2), 1647–1652 (2019).

[14] Y. Di, Y. Zhang, Y. Wen, H. Yao, Z. Zhou, Z. Ren and J. Shao, "Inchworm-inspired soft robot with controllable locomotion based on self-sensing of deformation," *IEEE Rob. Autom. Lett.* **9**(5), 4345–4352 (2024).

[15] A. Dumlu and K. Erenturk, "Trajectory tracking control for a 3-dof parallel manipulator using fractional-order PIλDμ control," *IEEE Trans. Ind. Electron.* **61**(7), 3417–3426 (2013).

[16] R. Mondal and J. Dey, "Performance analysis and implementation of fractional order 2-DOF control on cart-inverted pendulum system," *IEEE Trans. Ind. Appl.* **56**(6), 7055–7066 (2020).

[17] X. Ni, Y. Wei, S. Zhou and M. Tao, "Multi-objective network resource allocation method based on fractional PID control," *Signal Process.* **227**, 109717 (2025).

[18] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy and R. Vasudevan, "Data-driven control of soft robots using Koopman operator theory," *IEEE Trans. Rob.* **37**(3), 948–961 (2020).

[19] M. Švec, Š. Ileš and J. Matuško, "Predictive direct yaw moment control based on the koopman operator," *IEEE Trans. Control Syst. Technol.* **31**(6), 2912–2919 (2023).

[20] H. Shi and M. Q. H. Meng, "Deep Koopman operator with control for nonlinear systems," *IEEE Rob. Autom. Lett.* **7**(3), 7700–7707 (2022).

[21] M. Korda and I. Mezić, "Optimal construction of Koopman eigenfunctions for prediction and control," *IEEE Trans. Autom. Control* **65**(12), 5114–5129 (2020).

[22] M. S. Ramadan and M. Anitescu, "Extended Kalman filter-Koopman operator for tractable stochastic optimal control," *IEEE Control Syst. Lett.* **8**, 1643–1648 (2024).

[23] H. Wang, W. Liang, B. Liang, H. Ren, Z. Du and Y. Wu, "Robust position control of a continuum manipulator based on selective approach and Koopman operator," *IEEE Trans. Ind. Electron.* **70**(12), 12522–12532 (2023).

[24] G. Mamakoukas, M. L. Castano, X. Tan and T. D. Murphey, "Derivative-based Koopman operators for real-time control of robotic systems," *IEEE Trans. Rob.* **37**(6), 2173–2192 (2021).

[25] Z. Li, M. Han, D. N. Vo and X. Yin, "Machine learning-based input-augmented Koopman modeling and predictive control of nonlinear processes," *Comput. Chem. Eng.* **191**, 108854 (2024).

[26] S. Iman and M. R. Jahed-Motlagh, "Data-driven predictor of control-affine nonlinear dynamics: Finite discrete-time bilinear approximation of koopman operator," *Appl. Math. Comput.* **487**, 129068 (2025).

[27] M. Rahmani and S. Redkar, "Deep neural data-driven Koopman fractional control of a worm robot," *Expert Syst. Appl.* **256**, 124916 (2024).

[28] R. Mao, T. Meng, K. Wang, J. Lei and W. Wang, "Deep Koopman-operator-based model predictive control for free-floating space robots with disturbance observer," *Aerospace Sci. Technol.* **154**, 109515 (2024).

[29] A. T. Mallen, H. Lange and J. N. Kutz, "Deep probabilistic Koopman: Long-term time-series forecasting under periodic uncertainties," *Int. J. Forecasting* **40**(3), 859–868 (2024).

[30] Q. Peng, G. Chen, J. Guo and Z. Guo, "Attitude/orbit engine compound control for hypersonic target interceptor with discrete-time inputs and state constraints," *IEEE Trans. Aerospace Electron. Syst.* **60**(4), 4939–4951 (2024).

[31] Y. Liu, H. Wang, B. Liu, L. Chen, Y. Wang and H. Chen, "Learning-based compound docking control for UAV aerial recovery: Methodology and implementation," *IEEE/ASME Trans. Mechatron.* **28**(3), 1706–1717 (2022).

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press

[32] X. Zhou, W. Wang and Y. Shi, "Neural network/PID adaptive compound control based on RBFNN identification modeling for an aerial inertially stabilized platform," *IEEE Trans. Ind. Electron.* **71**(12), 16514–16522 (2024).
[33] A. Bounemeur, M. chemachema, A. Zahaf and S. Bououden, "Adaptive Fuzzy Fault-Tolerant Control Using Nussbaum Gain for a Class of SISO Nonlinear Systems with Unknown Directions," *Proceedings of the 4th International Conference on Electrical Engineering and Control Applications: ICEECA 2019, 17–19 December 2019, Constantine, Algeria* (Springer Singapore, 2019) pp. 493–510.
[34] A. Bounemeur and M. Chemachema, "Adaptive fuzzy fault-tolerant control using Nussbaum-type function with state-dependent actuator failures," *Neural Comput. Appl.* **33**(1), 191–208 (2021).
[35] O. Bey and M. Chemachema, "Finite-time event-triggered output-feedback adaptive decentralized echo-state network fault-tolerant control for interconnected pure-feedback nonlinear systems with input saturation and external disturbances: A fuzzy control-error approach," *Inf. Sci.* **669**, 120557 (2024).
[36] A. Bounemeur and M. Chemachema, "Finite-time output-feedback fault tolerant adaptive fuzzy control framework for a class of MIMO saturated nonlinear systems," *Int. J. Syst. Sci.* 56(4), 733–752 (2025).
[37] Z. Liu, A. M. Dizqah, J. M. Herreros, J. Schaub and O. C. Haas, "A hierarchical economic model predictive controller that exploits look-ahead information of roads to boost engine performance," *IEEE Trans. Control Syst. Technol.* **31**(6), 2632–2643 (2023).
[38] Y. Ma, T. Qin and Y. Li, "Nonlinear extended state observer based super-twisting terminal sliding mode synchronous control for parallel drive systems," *IEEE/ASME Trans. Mechatron.* **28**(6), 3087–3098 (2023).
[39] J. Baltes, G. Christmann and S. Saeedvand, "A deep reinforcement learning algorithm to control a two-wheeled scooter with a humanoid robot," *Eng. Appl. Artif. Intell.* **126**, 106941 (2023).
[40] H. Sekkat, O. Moutik, B. E. Kari, Y. Chaibi, T. A. Tchakoucht and A. E. H. Alaoui, "Beyond simulation: Unlocking the frontiers of humanoid robot capability and intelligence with Pepper's open-source digital twin," *Heliyon* **10**(14), e34456 (2024).
[41] M. Rojas, G. Hermosilla, D. Yunge and G. Farias, "An easy to use deep reinforcement learning library for AI mobile robots in Isaac sim," *Appl. Sci.* **12**(17), 8429 (2022).
[42] B. Hondow, A tiny green inchworm creeping along a pine needle, Dreamstime (n.d.). Available: https://www.dreamstime.com/green-inchworm-looping-its-way-along-pine-needle-little-creatures-look-like-caterpillars-not-larvae-image108904639.
[43] A. Ghanbari and S. M. R. S. Noorani, "Optimal trajectory planning for design of a crawling gait in a robot using genetic algorithm," *Int. J. Adv. Rob. Syst.* **8**(1), 6 (2011).
[44] M. R. Islam, M. Rahmani and M. H. Rahman, "A novel exoskeleton with fractional sliding mode control for upper limb rehabilitation," *Robotica* **38**(11), 2099–2120 (2020).

Cite this article: M. Rahmani, J. B. Menon, A. R. Uplap and S. Redkar (2025). "Compound Koopman data-driven control for an inchworm robot: validation through virtual experiments", Robotica **43**, 1591–1607. https://doi.org/10.1017/S0263574725000578

https://doi.org/10.1017/S0263574725000578 Published online by Cambridge University Press