Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Machine Learning-Based Nonlinear Control of a Ball and Beam System Driven by an Elastic Actuator

Sanam Hajipour

hajipour@b-tu.de

Brandenburg University of Technology Cottbus–Senftenberg

Dieter Bestle

Brandenburg University of Technology Cottbus–Senftenberg

Research Article

Keywords: Ball and beam system, Sliding mode control, Data-driven control, Underactuated system

Posted Date: June 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9888953/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

The 8th International Conference on Multibody System Dynamics
June 16-19, 2026, Seville, Spain

Machine Learning-Based Nonlinear Control of a Ball and Beam System Driven by an Elastic Actuator

S. Hajipour*, D. Bestle #

* Chair of Engineering Mechanics and Vibrations
Brandenburg University of Technology Cottbus-Senftenberg
03046 Cottbus, Germany
Hajipour@b-tu.de

University professor retired from Chair of Engineering Mechanics and Vehicle Dynamics
Brandenburg University of Technology Cottbus-Senftenberg
03046 Cottbus, Germany
Bestle@b-tu.de

ABSTRACT

Control of a ball rolling on a rigid beam has long served as a benchmark problem for nonlinear control of underactuated systems. The system poses significant challenges due to its coupled underactuated dynamics: the ball's motion can't be controlled directly but depends on the beam angle, while the beam is driven by an actuator. When an elastic joint is introduced to the actuator, the dynamics becomes even more challenging by introducing additional vibration modes. Control design has to ensure accurate tracking of the ball, robustness and stability under such conditions. The present contribution explores a hybrid tracking control strategy integrating inverse dynamics modeling with a nonlinear feedback controller. Based on simulation results, a data-driven surrogate model for the system's inverse dynamics is trained to approximate the nonlinear mapping between desired system motion and the required control input. This model predicts a feedforward action that generates the force needed to follow a path of both ball and beam, such that the ball moves from an initial rest position to a final rest one in finite time. Controller parameters are tuned through a genetic optimization algorithm to balance tracking performance and control smoothness. However, because learned models inevitably contain approximation errors, an additional robust feedback loop has to be defined. For the feedback loop, the super-twisting sliding mode control algorithm is introduced as additive correction to the feedforward term. Overall, the combined strategy demonstrates how modern learning-based models and robust nonlinear control can be effectively integrated to handle complex underactuated multibody systems.

Keywords: Ball and beam system, Sliding mode control, Data-driven control, Underactuated system.

1 INTRODUCTION

Ball and beam systems are a canonical benchmark for underactuated nonlinear control because a single actuator regulates the beam angle while the ball position evolves through coupled, unstable dynamics. Their strong nonlinearity, sensitivity to parameter variations, and strict performance demands in trajectory tracking make them a frequent testbed for both classical and learning-based control design [8, 1]. Recent work spans a wide range from fully model-based control to purely model-free reinforcement learning (RL) with multiple studies explicitly benchmarking these paradigms. Several studies report that machine learning controllers can reach competitive tracking accuracy, but classical controllers often retain advantages in transient performance. In [15] it is shown that a DQN agent achieves competitive position accuracy, yet a conventional proportional-integral-derivative (PID) controller produces a superior settling time of 1.3sec. In [23] it is found that a policy model linearization (PML) approach outperforms PID and fuzzy control (FC) with a performance comparable to linear quadratic regulator (LQR) though with longer regulation time. In model-based optimization, [12] reports that GA-tuned (Genetic Algorithm) LQR yields the lowest steady-state error among tested approaches, outperforming model-free PID and hybrid PID-LQR designs. Likewise, [22] demonstrates a neural network-based controller with performance comparable to linear quadratic Gaussian (LQG), while improving generalization across reference signals. Collectively, these results indicate that while data-driven methods can match or exceed classical control designs under certain conditions, performance depends strongly on the structure embedded in the controller and the quality of the optimization objective.

A key differentiator across this literature is the degree to which the controller relies on explicit physics. In [12] a model-based approach achieves the lowest steady-state error but requires detailed Lagrangian modeling including friction, rotary inertia, and Coriolis terms. At the other end, model-free RL approaches (PPO, DDPG, SAC, TD3) avoid explicit modeling but typically demand substantial training data and interaction [3, 8, 23]. Hybrid methods attempt to balance these trade-offs: [22] incorporates stability-related constraints on a neural network controller to achieve LQG-comparable performance. Further [23] highlights how embedding domain knowledge into the reward function can reduce training burden, achieving learning in 120 episodes with 500 learning steps by encoding control objectives directly into the learning signal. This supports the following conclusions: model-based designs reduce data needs at the cost of accurate characterization; model-free designs tolerate unknown dynamics at the cost of extensive interaction; and hybrid designs offer intermediate performance by injecting structure while retaining adaptability[23, 22, 1, 20].

In parallel, sliding mode control (SMC) has been repeatedly validated as a robust alternative for ball and beam stabilization and tracking. Robust SMC is implemented to stabilize ball and beam systems by varying the beam angle [21], and experimental comparisons [4] show that SMC can outperform PI control in tracking, response time and accuracy by cascaded architectures. Nonlinear SMC combined with feedback linearization improves tracking compared to linear PD in simulation [5] , while observer-based SMC frameworks are validated experimentally when velocity sensing is unavailable [13]. Additional variants include approximate feedback-linearized SMC derived from Lagrange modeling [7], dynamic sliding mode control (DSMC) for disturbance rejection and chattering reduction [14], adaptive hierarchical SMC with Lyapunov-stability guarantees [18], and second-order SMC approaches supported by theory, simulation, and experiments [24]. They collectively demonstrate that SMC-family methods offer strong robustness and implementability. Hybrid SMC enhancements using learning components (e.g., RBF networks and fuzzy logic) report large transient and overshoot improvements [11], and combined SMC-fuzzy-MLP neural designs are proposed for robustness under uncertainty [9]. However, despite the maturity of SMC variants in this domain, the reviewed SMC studies primarily assume conventional rigid actuation.

A recurring gap emerges when actuator technology is considered. The underactuated, unstable and nonlinear nature of the ball and beam system creates fundamental control challenges that conventional approaches can struggle with [19]. Yet all reviewed studies employ conventional actuators (servo, stepper and DC motors) [6, 15]. The absence of elastic actuators, whose compliance and energy storage properties alter the closed-loop behavior, represents a significant gap relative to the stated research question. This suggests that control strategies, including machine learning (ML) and robust nonlinear controllers, remain underexplored for ball and beam systems with elastically driven dynamics, where actuator compliance introduces bandwidth limits and potential resonance effects that standard rigid-actuation assumptions do not capture.

The present study's strength lies in its computational control framework that explicitly targets robust tracking while remaining practical under modeling imperfections. The approach combines (i) a physics-informed feedforward term computed via inverse dynamics (including a surrogate option when the full inverse model is unavailable or costly), and (ii) a second-order sliding mode feedback law (super-twisting) to enforce robustness. Controller parameters are tuned through optimization using a simulation-in-the-loop cost that can incorporate tracking and smoothness objectives, enabling systematic balancing of regulation quality and chattering behavior. The framework is positioned to extend to elastically actuated variants of the ball and beam system, addressing the actuator-related gap identified in the literature while retaining robustness benefits established by the SMC-family [4, 24, 9] and the performance-driven tuning benefits demonstrated in GA-based controller optimization studies [12].

The remainder of this paper is organized as follows: Section 2 presents the system modeling including the elastic-joint actuation concept and the resulting equations of motion. Section 3.1 introduces the open-loop feedforward strategy based on inverse dynamics and its surrogate implementation. The proposed closed-loop feedback controller based on the super-twisting sliding mode algorithm is described in Section 4, where Section 4.2 outlines the parameter tuning procedure. Simulation results and their discussion are reported in Section 5, before Section 6 concludes the proposed method.

2 SYSTEM MODELING

In this paper, the planar ball and beam system in Fig. 1 with an elastic joint is considered. A homogeneous rigid beam (mass M, length L, moment of inertia I_c w.r.t. center of gravity C) rotates in the vertical x,$y-$plane about a pivot fixed to the inertial frame, where the beam angle $\varphi(t)$ is measured with respect to the horizontal axis. A homogeneous ball (mass m, diameter D) rolls on the beam without slipping where the generalized coordinate $r(t)$ denotes the ball position measured along the beam. The control input is denoted by $\psi(t)$ acting on the beam through a compliant element of stiffness k at the pivot. The system parameters are listed in Table 1, where all units are given in SI-units. According to Fig. 1c the ball is rolling in a 90°-groove which is modeled through an effective rolling radius, $r_B = D/(2\sqrt{2})$. The ball inertia w.r.t. its center of gravity can be written in compact form as $I_B =: \frac{2}{5}m(D/2)^2 =: \mu\, m r_B^2$ where $\mu = 4/5$.

Fig. 1b shows a free body diagram and the kinematics of the system, which can be described by the generalized coordinates

$$\boldsymbol{q}(t) = \begin{bmatrix} r(t) & \varphi(t) \end{bmatrix}^\top. \tag{1}$$

The equations of motion may be derived from Newton's and Euler's laws using the Newton-Euler formalism for holonomic multibody systems resulting in

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{q^e}(\boldsymbol{q}) = \boldsymbol{h}\,\psi, \tag{2}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}})$ collects velocity-dependent terms, $\boldsymbol{q^e}(\boldsymbol{q})$ collects gravity and elastic effects, and $\boldsymbol{h}$ is the input vector where the input ψ acts on the beam via the moment $M_o = k(\psi - \varphi)$. Expanding (2) yields the explicit model used for simulation:

$$\boldsymbol{M}(\boldsymbol{q}) \begin{bmatrix} \ddot{r} \\ \ddot{\varphi} \end{bmatrix} + \begin{bmatrix} -mr\,\dot{\varphi}^2 \\ 2mr\,\dot{r}\,\dot{\varphi} \end{bmatrix} = \begin{bmatrix} -mg\sin\varphi \\ -\left(\frac{ML}{2} + mr\right) g\cos\varphi + mgr_B\sin\varphi \end{bmatrix} + \begin{bmatrix} 0 \\ k \end{bmatrix} (\psi - \varphi), \tag{3}$$

where the inertia matrix is

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} (1+\mu)m & -(1+\mu)mr_B \\ -(1+\mu)mr_B & \frac{ML^2}{3} + (1+\mu)mr_B^2 + mr^2 \end{bmatrix}. \tag{4}$$

The state vector

$$\boldsymbol{x}(t) = \begin{bmatrix} r(t) & \varphi(t) & \dot{r}(t) & \dot{\varphi}(t) \end{bmatrix}^\top \tag{5}$$

is influenced by the control input $\psi(t)$, where the control objective is trajectory tracking for both ball position and beam angle. Because the input does not directly act on the ball coordinate, Eq.(2) cannot be inverted for ψ as function of the desired ball motion from a single equation. Instead, ψ must be derived from the beam equation together with the desired ball motion (inverse dynamics), as discussed in the next section.

Table 1: System parameters(numbers given in SI units).

beam					ball			
M	L	r_c	I_c	k	m	D	I_B	r_B
0.15	0.5	$L/2$	$ML^2/12$	10	0.26	0.04	$\mu m r_B^2$	$D/2\sqrt{2}$

3 OPEN-LOOP FEEDFORWARD CONTROL

The goal of the experiment is to move the ball from initial rest position $(r,\dot{r})\big|_{t=0} = (0,0)$ to a final rest position $(r,\dot{r})\big|_{t=T} = (r_c,0)$ within a given time T. This may be achieved by lowering the beam in a first phase I to accelerate the ball, and then rising it to decelerate the ball in a second phase II as shown in Fig. 2. In order to fulfill the two final conditions, two parameters t_m and $\dot{\varphi}_m$ may be introduced connecting the two phases at $\varphi(t_m) = 0$. In order to fulfill equilibrium conditions at the beginning and end of the maneuver,

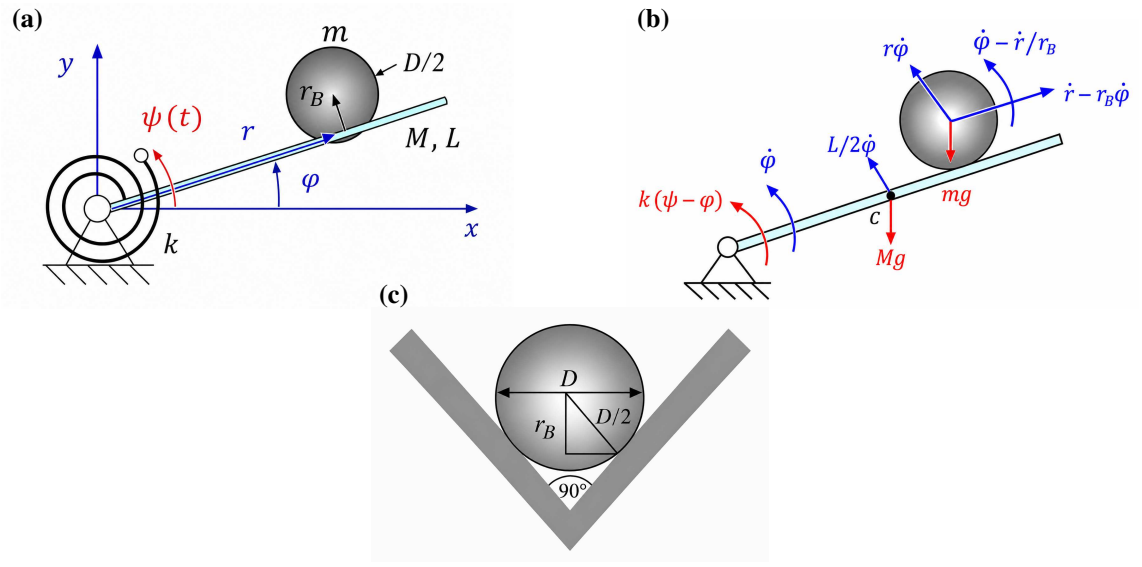


Fig. 1. Ball and beam system (a), with kinematics and free-body diagram (b) and rolling conditions (c)

angles and angular velocities have to vanish. Additionally, the beam acceleration should be zero at the beginning, end and connection point to avoid velocity jumps. All these conditions may be summarized as follows:

$$(\varphi, \dot{\varphi}, \ddot{\varphi})_{t=0} = \mathbf{0}, \quad (\varphi, \dot{\varphi}, \ddot{\varphi})_{t=T} = \mathbf{0}, \quad (\varphi, \ddot{\varphi})_{t=t_m} = \mathbf{0}, \quad \dot{\varphi}(t_m) = \dot{\varphi}_m. \tag{6}$$

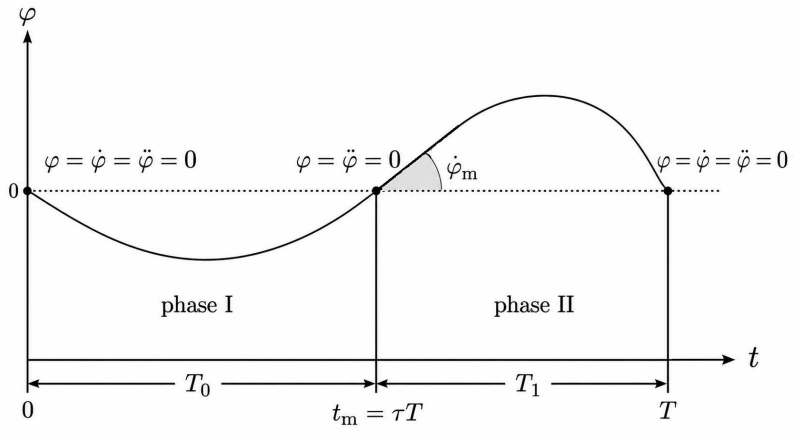


Fig. 2. Desired beam motion

3.1 Desired beam motion

These requiremens may be fulfilled by combining two 5th-order polynomials as

$$\varphi := \varphi_d(t) = \begin{cases} H(\zeta)\, T_0\, \dot{\varphi}_m & \text{for phase I:} \quad 0 \le t \le t_m \quad T_0 := t_m, \quad \zeta = \frac{t}{T_0} \\ -H(\zeta)\, T_1\, \dot{\varphi}_m, & \text{for phase II:} \quad t_m \le t \le T, \quad T_1 = T - t_m, \quad \zeta = \frac{(T-t)}{T_1} \end{cases} \tag{7}$$

with some basis function $H(\zeta)$, $0 \le \zeta \le 1$. Due to

$$\dot{\varphi}_d = H'\dot{\varphi}_m, \qquad \ddot{\varphi}_{d,\mathrm{I}} = \frac{H''\dot{\varphi}_m}{T_0}, \qquad \ddot{\varphi}_{d,\mathrm{II}} = -\frac{H''\dot{\varphi}_m}{T_1}, \tag{8}$$

conditions (6) transfer to the basis function $H(\zeta)$ as

$$H(0) = H'(0) = H''(0) = 0, \qquad H(1) = H''(1) = 0, \qquad H'(1) = 1. \tag{9}$$

These six conditions are satisfied by the 5th-order polynomial

$$H(\zeta) = \zeta^3 \left(-4 + 7\zeta - 3\zeta^2\right). \tag{10}$$

The two parameters τ (determining $t_m = \tau T$) and $\dot{\varphi}_m$ have a nonlinear effect, which is why they need to be found by solving the nonlinear optimization problem

$$\min_{\tau, \dot{\varphi}_m} \left[\left(r(T) - r_c\right)^2 + \dot{r}^{\,2}(T) \right] \tag{11}$$

e.g. resulting from the terminal conditions. This can be achieved, by a SQP-algorithm (Sequential Quadratic Programming), where $r(T)$ and $\dot{r}(T)$ have to be computed by simulating the first equation of Eq. (3). Substitution of $\varphi(t) := \varphi_d(t; \tau, \dot{\varphi}_m)$ simplifies the ball equation to

$$\ddot{r}_d(t) = r_B\, \ddot{\varphi}_d(t) + \frac{r_d(t)\, \dot{\varphi}_d^2(t) - g \sin\!\big(\varphi_d(t)\big)}{1 + \mu}. \tag{12}$$

As a result, we find for $T = 3$ the solution $\tau = 0.5$, $\dot{\varphi}_m = 0.1803$. The complete information on the open-loop control is then given by Eq. (7) and Eq. (8) for the beam and Eq. (12) and its integrals for the ball. It may be summarized by the extended state

$$\boldsymbol{x}_d(t) = \left[r_d(t) \quad \varphi_d(t) \quad \dot{r}_d(t) \quad \dot{\varphi}_d(t) \quad \ddot{r}_d(t) \quad \ddot{\varphi}_d(t) \right]^\top. \tag{13}$$

3.2 Inverse-dynamics feedforward input

The nominal open-loop input $\psi_{\text{ff}}(t)$ may be computed from inverse dynamics using $\boldsymbol{x}_d(t)$. It may be obtained either from analytic inverse dynamics or from a learned surrogate model. The resulting $\psi_{\text{ff}}(t)$ is then applied to the plant model in open loop to generate the responses $r(t)$ and $\varphi(t)$. Figure 3 summarizes the open-loop structure used in this work starting from the desired beam motion $\varphi_d(t)$ and its derivatives. These signals drive the ball-dynamics propagation to obtain $\ddot{r}_d(t)$ and integrals $r_d(t)$, $\dot{r}_d(t)$. The inverse-dynamics block (analytic or surrogate) finally computes $\psi_{\text{ff}}(t)$, being applied to the plant.

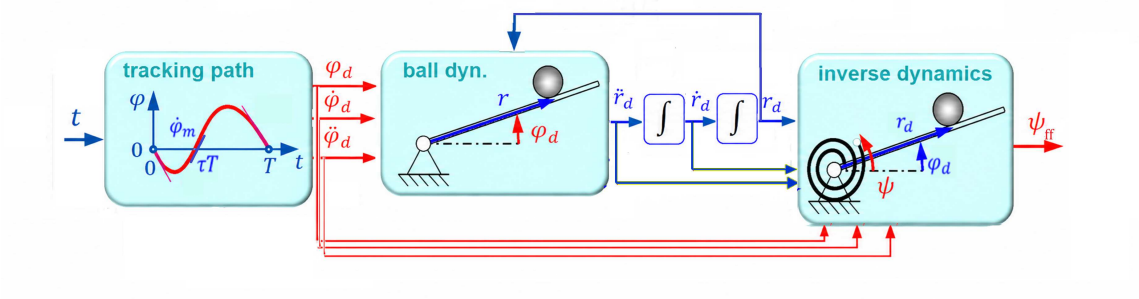


Fig. 3. Open-loop feedforward structure

In simple cases, the inverse dynamics

$$\psi_{\text{ff}}(t) = \mathcal{I}(\boldsymbol{x}_d(t)) \tag{14}$$

may be obtained analytically, otherwise a data-driven surrogate model

$$\hat{\psi}_{\text{ff}}(t) = \hat{\mathcal{I}}(\boldsymbol{x}_d(t); \boldsymbol{\vartheta}) \approx \psi_{\text{ff}}(t) \tag{15}$$

may replace Eq.(14). In the latter case, the parameter vector $\boldsymbol{\vartheta}$ is learned from a dataset generated by simulation of the forward dynamics. Training samples are obtained by exciting the plant with any smooth time-varying inputs $\psi^{(i)}(t)$ and recording the resulting motion $\boldsymbol{q}^{(i)}(t)$. Using central differences for differentiation, corresponding derivatives $\left(\dot{\boldsymbol{q}}^{(i)}(t), \ddot{\boldsymbol{q}}^{(i)}(t)\right)$ are computed and paired with the applied input $\psi^{(i)}(t)$, yielding a dataset [2]

$$\mathcal{D} = \left\{ \left(\boldsymbol{q}_j{}^{(i)}, \dot{\boldsymbol{q}}_j{}^{(i)}, \ddot{\boldsymbol{q}}_j{}^{(i)}; \psi_j^{(i)} \right), i = 1 \ldots N \right\}, \tag{16}$$

where index j refers to the recorded time points t_j and $\bullet_j = \bullet(t_j)$. Here, the underlying simulation starts from random initial conditions and uses constant random values for input $\psi^{(i)}$.

A multilayer perceptron architecture with two hidden layers containing 20 and 10 neurons, respectively, was implemented using MATLAB's standard fitnet framework with $tanh$ activation function. The dataset was divided into training and testing samples by ratio (85:15). The surrogate is trained for 2000 epochs using mean squared error minimization. After training, the network is exported to provide a fast, purely data-driven approximation (15) of the feedforward input for open-loop computations.

In the present simple case, the exact inverse dynamics (14) can be found from the second equation of the equations of motion (3) as

$$\begin{aligned}\psi_{\mathrm{ff}} = \varphi_d + \frac{1}{k}\Big(&- (1+\mu) m r_B \ddot{r}_d + \Big(\frac{ML^2}{3} + (1+\mu) m r_B^2 + m r_d^2\Big)\ddot{\varphi}_d \\ &+ 2 m r_d \dot{r}_d \dot{\varphi}_d + \Big(\frac{ML}{2} + m r_d\Big) g \cos\varphi_d - m g r_B \sin\varphi_d\Big)\end{aligned} \tag{17}$$

and used for assessing the quality of the surrogate. Figure 4 shows an almost perfect agreement between the surrogate predictions with this analytical result indicating an accurate approximation of the inverse dynamics function. The surrogate model achieves a root mean squared error of $\mathrm{RMSE}_\psi = 1.14 \times 10^{-5}$ over all evaluated trajectories where Fig. 5 shows the distribution of the data used for the surrogate training in the position, velocity, and acceleration spaces. The generated trajectories cover a wide range of operating conditions of the system, including both slow and fast as well as positive and negative motions. This broad coverage helps the neural network to learn the inverse dynamics over the relevant operating region and reduces the likelihood of extrapolation during closed-loop operation. Consequently, the surrogate model can provide reliable predictions for the trajectories encountered in the control simulations.

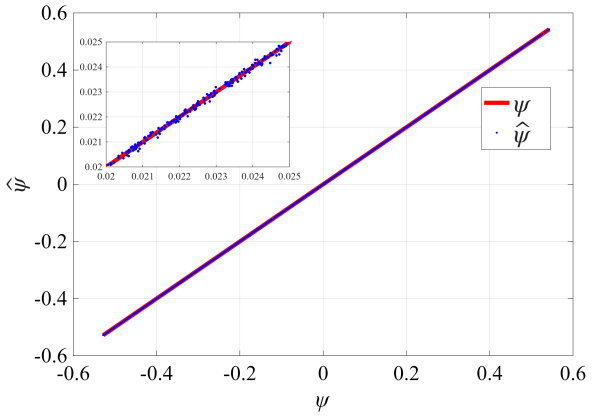


Fig. 4. Surrogate prediction of the inverse dynamics control input

4 CLOSED-LOOP FEEDBACK CONTROL

Although feedforward input $\hat{\psi}_{\mathrm{ff}}(t)$ provides an almost perfect behavior, its performance is sensitive to modeling errors and measurement noise. Therefore, it has to be augmented by a closed-loop controller to ensure robust tracking of the prescribed references for the beam angle and ball position, while maintaining stability of the nonlinear underactuated system.

4.1 Super-twisting sliding mode control (STSMC)

A nonlinear feedback action is added to compensate the residual dynamics that is not captured by the feedforward model. The applied control input is formed by combining the inverse-dynamics feedforward

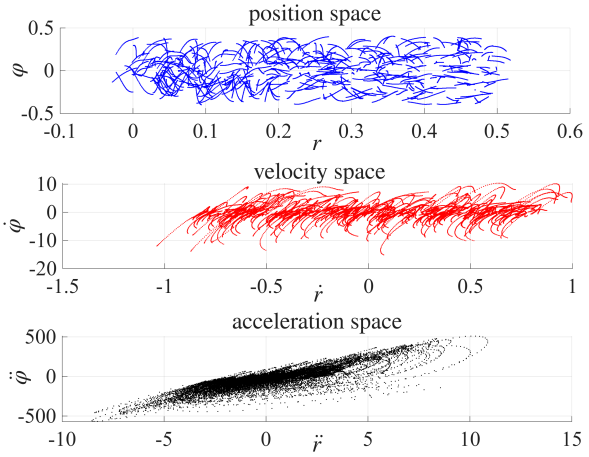


Fig. 5. Coverage of the surrogate-training dataset in the position, velocity, and acceleration spaces

term (15) with an additive feedback correction, i.e.,

$$\psi(t) = \hat{\psi}_{\mathrm{ff}}(t) + \Delta\psi(t), \tag{18}$$

where $\Delta\psi(t)$ is designed to enforce convergence to the desired motion. This architecture preserves the accuracy of model-based feedforward when the model is correct, while the feedback term provides robustness when discrepancies occur. To generate $\Delta\psi(t)$, a second-order sliding mode controller based on the super-twisting algorithm is used [16]. Super-twisting sliding mode control is well suited for this problem because it provides finite-time convergence of a properly defined sliding variable while reducing the chattering typically associated with first-order sliding mode control through a continuous control signal.

If the plant state (5) differs from the reference trajectories $\varphi_d(t)$, $r_d(t)$, this results in position and velocity tracking errors

$$e_\varphi(t) = \varphi(t) - \varphi_d(t), \qquad \dot{e}_\varphi(t) = \dot{\varphi}(t) - \dot{\varphi}_d(t), \tag{19}$$

$$e_r(t) = r(t) - r_d(t), \qquad \dot{e}_r(t) = \dot{r}(t) - \dot{r}_d(t). \tag{20}$$

Two separate first-order sliding surfaces are chosen for beam angle and ball motion errors as

$$s_1(t) = \dot{e}_\varphi(t) + \lambda_1 e_\varphi(t), \tag{21}$$

$$s_2(t) = \dot{e}_r(t) + \lambda_2 e_r(t) \tag{22}$$

with control design parameters $\lambda_1, \lambda_2 > 0$. The feedback correction $\Delta\psi(t)$ is generated from a combined sliding variable

$$s(t) = \lambda_3 s_1(t) + s_2(t) \tag{23}$$

using the super-twisting algorithm [10], where $\lambda_3 > 0$. To alleviate chattering, the usually taken discontinuous sign function is replaced by a boundary-layer saturation function

$$\mathrm{sat}(s) = \frac{s}{|s| + \varepsilon}, \qquad \varepsilon > 0. \tag{24}$$

The resulting control law is chosen as

$$\Delta\psi(t) = -K_1 \sqrt{|s(t)|}\,\mathrm{sat}\big(s(t)\big) + \sigma(t), \tag{25}$$

$$\dot{\sigma}(t) = -K_2\,\mathrm{sat}\big(s(t)\big), \tag{26}$$

where $K_1, K_2 > 0$ are feedback gains and $\sigma(t)$ is an internal super-twisting state. The total input $\psi(t)$ then combines the surrogate-based open-loop control $\hat{\psi}_{\mathrm{ff}}(t)$ with the closed-loop control (25) as indicated in Fig. 6.

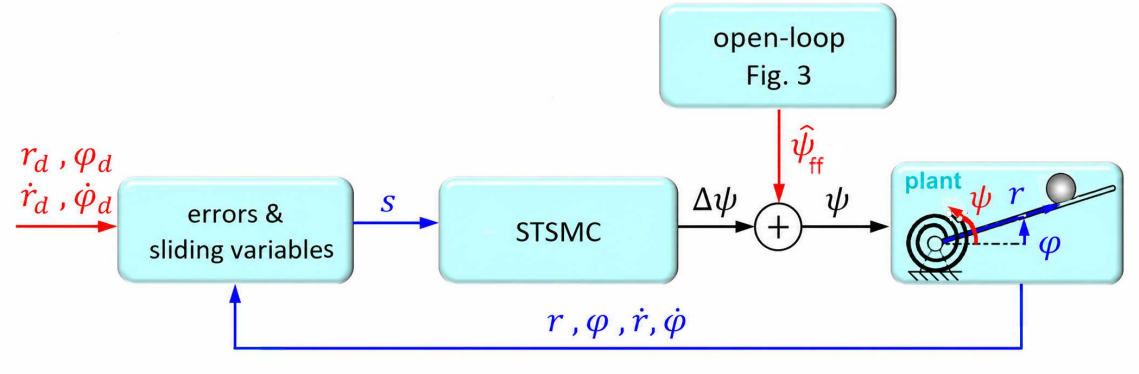


Fig. 6. Closed-loop feedback controller (STSMC)

4.2 PARAMETER TUNING USING A GENETIC ALGORITHM

The closed-loop controller contains several design parameters, namely the super-twisting gains (K_1, K_2) and the sliding-surface parameters $(\lambda_1, \lambda_2, \lambda_3)$ that strongly influence tracking performance and chattering. Since the resulting closed-loop dynamics is nonlinear and the performance indices are evaluated through numerical simulation, the tuning of these parameters is treated as a black-box optimization task. A genetic algorithm (GA) [17] is used to search for a suitable parameter vector

$$\boldsymbol{p} = \begin{bmatrix} K_1 & K_2 & \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix}^\top \in [\boldsymbol{p}_{\min}, \boldsymbol{p}_{\max}], \tag{27}$$

where $\boldsymbol{p}_{\min}$ and $\boldsymbol{p}_{\max}$ denote element-wise lower and upper bounds.

For each candidate $\boldsymbol{p}$ proposed by the GA, the nonlinear closed-loop system in Fig. 6 is simulated over a fixed time horizon $[0, T_f]$, $T_f \geq T$ and evaluated by a scalar cost function penalizing tracking errors of both beam and ball. This objective is defined as

$$J_{\text{trk}}(\boldsymbol{p}) = \frac{1}{T_f} \int_0^{T_f} \left[w\big(\varphi(t) - \varphi_d(t)\big)^2 + (1-w)\big(r(t) - r_d(t)\big)^2 \right] dt \tag{28}$$

with a user-defined weight $w \in [0, 1]$. To discourage chattering and excessive control activity, an additional smoothness term is included based on the variation of the applied input $\psi(t)$:

$$J_{\text{sm}}(\boldsymbol{p}) = \frac{1}{T_f} \int_0^{T_f} \Big(\dot{\psi}(t)\Big)^2 \, dt, \tag{29}$$

where in the discrete-time implementation the penalty is computed as a sum of step-to-step variations $\Delta\psi_k = \psi_k - \psi_{k-1}$. The overall GA objective is then formed as

$$J(\boldsymbol{p}) = J_{\text{trk}}(\boldsymbol{p}) + \alpha\, J_{\text{sm}}(\boldsymbol{p}), \tag{30}$$

where the user-defined weight $\alpha \geq 0$ controls the trade-off between tracking accuracy and control smoothness.

The GA starts from a randomly initialized population of candidate parameter vectors (27) within the bounds $[\boldsymbol{p}_{\min}, \boldsymbol{p}_{\max}]$. In each generation, candidates are evaluated via the cost function (30). The algorithm terminates after a predefined number of generations or when the improvement of the best objective value falls below a tolerance. The best individual returned by the GA is used as the tuned control parameter set for the closed-loop simulation reported in this work.

5 RESULTS AND DISCUSSION

This section summarizes the simulation results obtained for the ball and beam system driven by an elastic joint, and the proposed hybrid control scheme composed of feedforward and a super-twisting sliding mode feedback correction. The performance is evaluated in terms of tracking accuracy for both beam angle and ball position, as well as control behavior.

5.1 Open-loop feedforward performance

First, the open-loop results may be analyzed. Figure 7 compares the feedforward control $\psi_{\mathrm{ff}}(t)$ obtained from analytic inverse dynamics (17) with the surrogate prediction $\hat{\psi}_{\mathrm{ff}}(t)$ according to Eq.(15). A very close agreement is observed over the entire time interval which indicates that the surrogate captures the inverse relation with high accuracy and can reproduce the nominal control action required by the model.

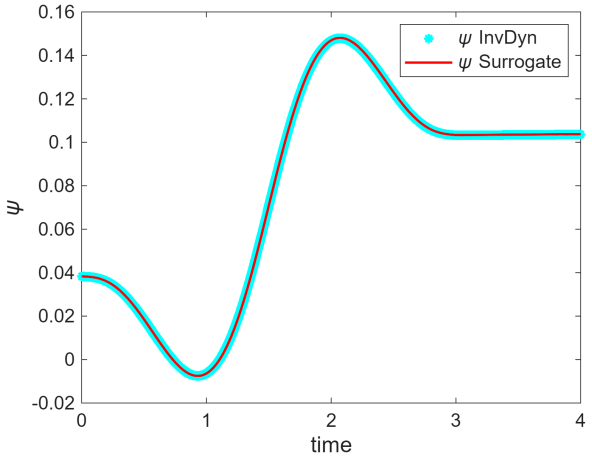


Fig. 7. Open-loop control

According to Fig. 8a, the ball position simulated for $\hat{\psi}_{\mathrm{ff}}(t)$ follows the desired trajectory (7) from the initial to the final rest position with almost no deviation. The ball motion is smooth and consistent with the rest-to-rest design objective. For the nominal model, the inverse dynamics feedforward input is obviously sufficient to generate the required beam motion for transporting the ball to the target location which is confirmed by the small ball position error with root mean squared error $\mathrm{RMSE}_{r,\mathrm{OL}} = 2.840 \times 10^{-4}$. The associated beam motion in Fig. 8b has an error of $\mathrm{RMSE}_{\varphi,\mathrm{OL}} = 1.118 \times 10^{-3}$. A close look to the end of the motion in Fig. 8a shows some drift-away of the ball position requiring an additional feedback loop for more robustness.

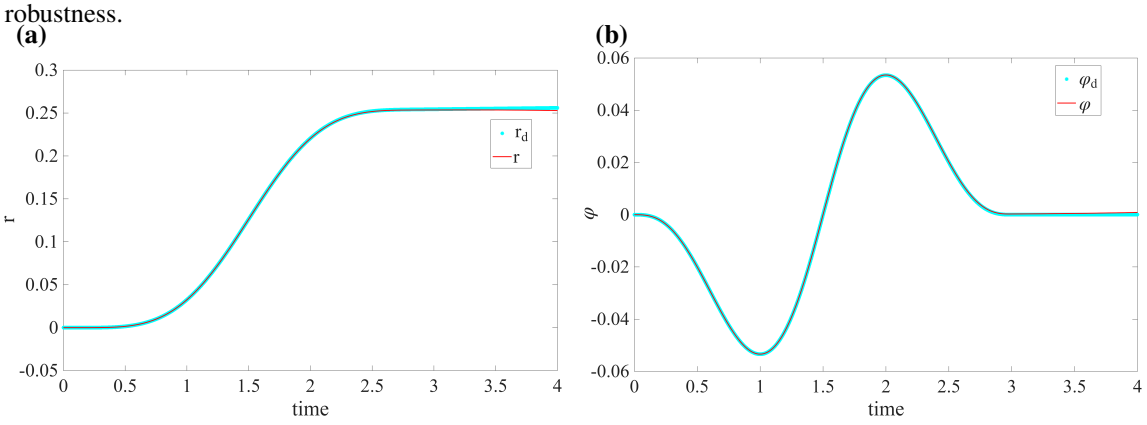


Fig. 8. Ball position (a) and beam angle (b) based on pure open-loop feedforward input

5.2 Closed-loop tracking with super-twisting feedback

Tuning the parameters (27) is critical to balancing fast convergence, robustness, and smooth control. Larger gains generally accelerate convergence but can increase control activity, while smaller gains reduce chattering but may slow down the response. The weighting parameter λ_3 sets the relative emphasis of beam versus ball tracking in the combined sliding variable (23); inappropriate weighting can lead to excellent tracking of one subsystem while allowing drift in the other. The controller parameters found by optimization for objective (28)-(30) with $T_f = 4$ and user-defined weight chosen by trial and error are listed in Table 2.

Table 2: Tuning of the closed-loop controller.

User-defined parameters			Optimized control parameters				
ε	α	w	K_1	K_2	λ_1	λ_2	λ_3
8×10^{-3}	0.1	0.5	0.214	4.891	9.604	0.175	0.997

Figure 9 shows the closed-loop tracking performance obtained with the super-twisting sliding mode controller. The beam angle $\varphi(t)$ follows the reference trajectory with an almost perfect overlap, yielding a root mean squared error of $\mathrm{RMSE}_{\varphi,CL} = 3.256 \times 10^{-8}$. The ball position $r(t)$ also tracks the desired motion $r_d(t)$ over the entire maneuver with $\mathrm{RMSE}_{r,CL} = 2.761 \times 10^{-7}$. From a sliding mode perspective, the super-twisting algorithm enforces convergence of the combined sliding variable $s(t)$ by shaping the error dynamics into a stable sliding regime. With the adopted boundary-layer saturation, exact finite-time convergence to $s = 0$ is replaced by practical convergence to an ε-dependent invariant set around the origin. Consequently, the tracking errors converge to a small bounded residual set whose size is governed by the chosen boundary-layer width providing a direct trade-off between chattering and steady-state accuracy.

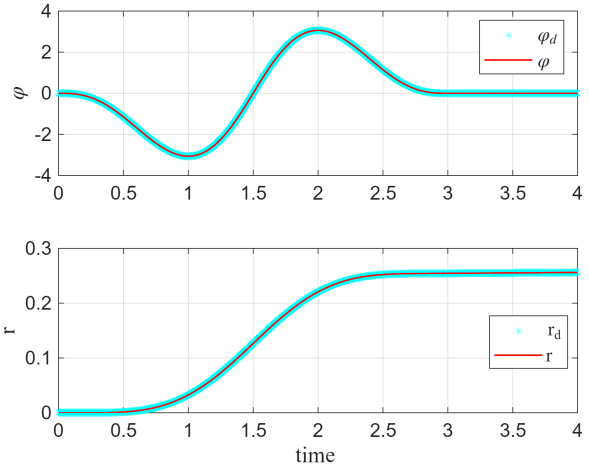


Fig. 9. Ball position and beam angle for closed-loop control

5.3 Error dynamics and phase portraits

The logarithmic plots of the absolute tracking errors provide a more detailed view of the closed-loop accuracy. As shown in Fig. 10a, $|e_r(t)|$ remains small throughout the simulation and increases only gradually to an the order of 10^{-6} near the end of the time interval. A similarly small magnitude is observed for the beam angle error $|e_\varphi(t)|$ in Fig. 10b, which stays below 10^{-7}. The corresponding phase portraits in Fig. 11 remain confined to a very small neighborhood of the origin. For the ball, the trajectory follows a smooth contraction toward $(e_r, \dot{e}_r) = (0, 0)$. The vertical peak close to $e_r \approx 0$ indicates a fast velocity correction in $\dot{e}_r$ while the position error is already close to zero, which can be related to the controller's final settling phase. For the beam, the loops are a characteristic signature of super-twisting sliding-mode regulation with a boundary-layer implementation. The control objective of STSMC is to drive the system toward the sliding manifold $s_2 = 0$; however, replacing $\mathrm{sign}(s_2)$ with the saturation yields *practical sliding*, i.e., convergence to a small tube around the manifold rather than exact sliding. Inside this tube, the feedback behaves as a continuous, high-gain nonlinear controller, continuously applying small corrective actions to keep $s_2(t)$ bounded. Moreover, the beam's second order dynamics in φ appears as small spiral trajectories in the $(e_\varphi, \dot{e}_\varphi)$ phase plane during the final phase. The size and density of the loops reflect the classical tradeoff in boundary-layer sliding control: a smaller ε (and/or larger gains $K_1, K_2, \lambda_1, \lambda_2, \lambda_3$) tightens tracking but increases high-frequency correction, while a larger ε reduces this activity at the cost of a larger residual error bound. Moreover, the internal super-twisting state σ together with the nonlinear $\sqrt{|s_2|}$ term can lead to slight over- and under-compensation near the origin.

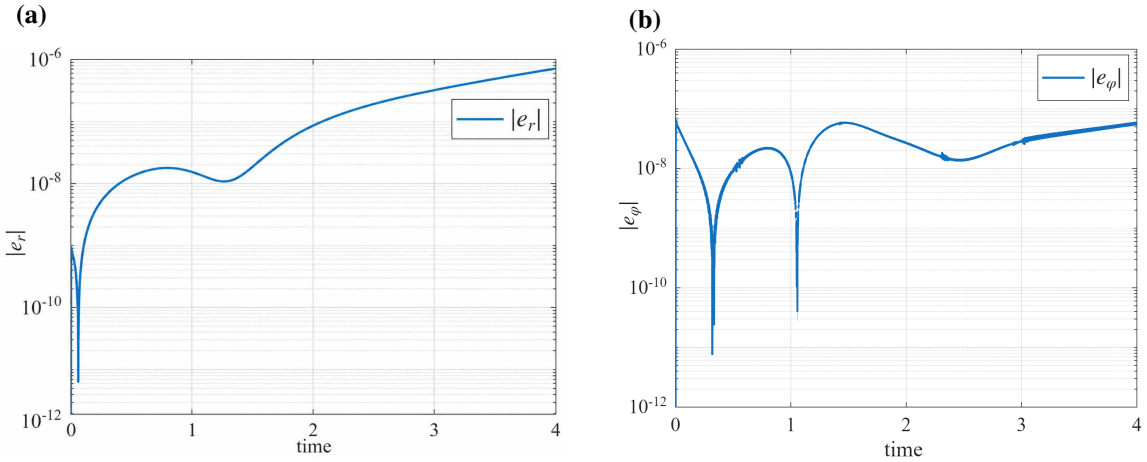


Fig. 10. Absolute ball (a) and beam (b) errors

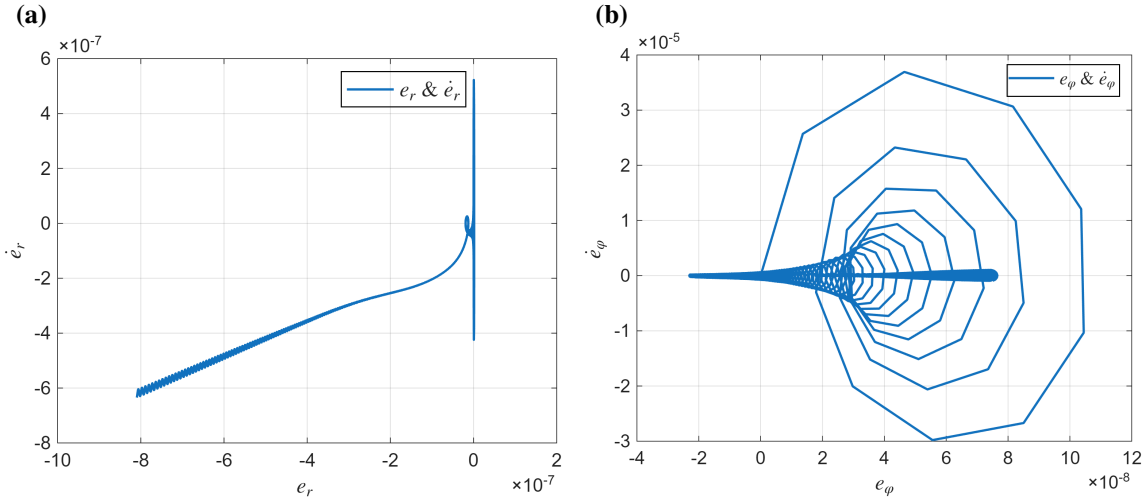


Fig. 11. Phase portraits of ball position (a) and beam angle (b) errors

5.4 Control performance

The applied control input $\psi(t)$ in Fig. 12a remains smooth and bounded and converges to a steady value as the system approaches the final equilibrium. The additive involved correction $\Delta\psi(t)$ in Fig. 12b is several orders of magnitude smaller than the total input, indicating that the feedforward term provides the dominant control action where as the super-twisting feedback mainly performs fine corrections to compensate remaining errors and enforce the sliding behavior. Moreover, $\Delta\psi(t)$ varies continuously without fast switching, which explains the absence of chattering in the applied input.

6 CONCLUSIONS

The investigated hybrid control strategy ia bout to solve the ball and beam benchmark problem even in the case of an elastic drive. The proposed control structure combines a feedforward term obtained from inverse dynamics with a nonlinear feedback law based on the super-twisting sliding mode algorithm. To reduce reliance on explicit analytical inversion, a data-driven surrogate of the inverse dynamics is trained from simulation data and utilized for feedforward computation. The results demonstrate that the inverse dynamics feedforward achieves accurate nominal tracking and that the surrogate reproduces the analytically computed feedforward input very closely. By adding super-twisting feedback correction, tracking performance is further enhanced while the applied control input remains smooth and bounded. The boundary layer implementation provides a practical means of reducing chattering while preserving high tracking accuracy. Control parameters are tuned by optimization to enable a systematic selection of gains for the nonlinear

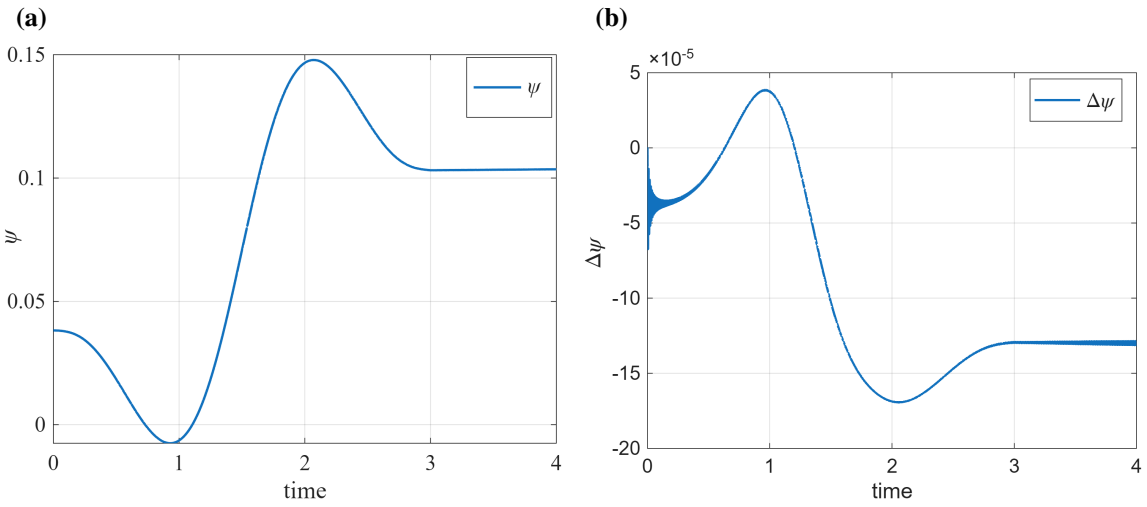


Fig. 12. Closed-loop control action (a) including sliding mode correction term (b)

closed-loop system balancing precision and chattering.

Future work may extend the evaluation to broader operating conditions and investigate the influence of modeling errors and pseudo-measurement noise.

Acknowledgments

This project was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the primal program SPP 2353 under project no. 501840485.

Statements and Declarations

The authors declare that they have no conflict of interest.

REFERENCES

[1] Amiruddin, B. P.; Kadir, R. E. A.: Ball and Beam Control Using Adaptive PID Based on Q-Learning. In: Proc. of 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI), pp. 203–208, IEEE,(2020).

[2] Bestle, D.; Hajipour, S.: Design of a Tracking Controller Based on Machine Learning. Int. J. of Mech. Sys. Dyn.5, 201–211(2025).

[3] Blake, X. J.; Aphiratsakun, N.: Reinforcement Learning for PID Gain Tuning in Ball-and-Beam Systems: A Comparative Study. In: Proc. of 1st International Conference on Robotics, Engineering, Science, and Technology (RESTCON), pp. 187–190, IEEE, (2024).

[4] Can, K.; Basci, A.: Position Control of a Ball & Beam Experimental Setup Based on Sliding Mode Controller. Int. J. of Appl. Math. Elec. and Comp.1, 29–35(2017).

[5] Cardona, S.; Ospina, S.; Giraldo, E.: Sliding Mode Control Based on Feedback Linearization for a Ball and Beam System. In: Proc. of 4th Colombian Conference on Automatic Control (CCAC), pp. 1–5, IEEE,(2019).

[6] Chenghai, Z.; Jianan, L.; Fudong, W.; Xinhao, Y.; Ying, Z.; Xiaoli, S.: Design and Implementation of Ball and Beam Control System Based on Iterative Learning Control. In: Proc. of Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 1288–1292, IEEE,(2016).

[7] Choukchou-Braham, A.; Bensalah, C.; Cherki, B.: Stabilization of an Under-Actuated Mechanical System by Sliding Mode Control. In: Proc. of AIP Conference, Vol. 1019, No. 1, pp. 80–84, American Institute of Physics,(2008).

[8] Ganesh, A.; Sundareswari, M. B.; Panda, M.; Mozhi, G. T.; Dhanalakshmi, K.: Reinforcement Learning Control of Servo Actuated Centrally Pivoted Ball on a Beam. In: Proc. of 15th International Conference on Industrial and Information Systems (ICIIS), pp. 103–108, IEEE,(2020).

[9] Hajipour, S.; Pourhashem, H.; Chegini, S. N.; Bagheri, A.: Optimized Neuro Observer-Based Sliding Mode Control for a Nonlinear System Using Fuzzy Static Sliding Surface. Appl. Soft Comp. 124, 108904(2022).

[10] Hajipour, S.; Zhai, T.; Bestle, D.: Data-Driven Sliding Mode Control of a Parallel 2D-Robot. In: Proc. of 8th International Conference on Intelligent Robotics and Control Engineering (IRCE), pp. 75–80, IEEE,(2025).

[11] Jitkhamheang, P.; Wongvanich, N.; Tangsrirat, W.: Design of RBF-Based Adaptive Gain Fuzzy Sliding Mode Control for Uncertain Ball and Beam System. In: Proc. of 9th International Conference on Business and Industrial Research (ICBIR), pp. 286–291, IEEE,(2024).

[12] Keshmiri, M.; Jahromi, A. F.; Mohebbi, A.; Amoozgar, M. H.; Wen-Fang, X.: Modeling and Control of Ball and Beam System Using Model Based and Non-Model Based Control Approaches. Int. J. on Smart Sens. and Intel. Sys. 5, 14(2012).

[13] Kim, J. S.; Park, G. M.; Choi, H. L.: Sliding Mode Control Design Under Partial State Feedback for Ball and Beam System. In: Proc. of International Conference on Control, Automation and Systems (ICCAS), pp. 1293–1296, IEEE,(2010).

[14] Khan, A. M.; Bhatti, A. I.; Khan, Q.: Static & Dynamic Sliding Mode Control of Ball and Beam System. In: Proc. of 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST), pp. 32–36, IEEE,(2012).

[15] Koczi, D.; Nemeth, J.; Sarosi, J.: Two-Dimensional Positioning with Machine Learning in Virtual and Real Environments. Elec. 12, 671(2023).

[16] Levant, A.: Sliding Order and Sliding Accuracy in Sliding Mode Control. Int. J. of Control. 58, 1247–1263(1993).

[17] Mirjalili, S.: Genetic Algorithm — Evolutionary Algorithms and Neural Networks. Studies in Computational Intelligence, Vol. 780. Cham: Springer,(2019).

[18] Qian, D.; Liu, X.; Li, L.: Adaptive Hierarchical Sliding Mode Control for Ball-Beam Systems. Int. J. of Adv. Mech. Sys. 4, 205–211(2012).

[19] Ravichandran, P.; Sathiyavathi, S.; Sathish Babu, S.; Vimala Starbino, A.: Hybrid Arrangement of Iterative Learning Control Strategy for Ball and Beam System. Ins. of Elec. and Tele. Eng. (IETE) Jour. of Res. 69, 916–923(2023).

[20] Romeres, D.; Jha, D. K.; DallaLibera, A.; Yerazunis, B.; Nikovski, D.: Semiparametrical Gaussian Processes Learning of Forward Dynamical Models for Navigating in a Circular Maze. In: Proc. of International Conference on Robotics and Automation (ICRA), pp. 3195–3202, IEEE,(2019).

[21] Soni, R.; Sathans: Robust Control of a Ball and Beam System Through Sliding Mode Controller. In: Proc. of International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR), pp. 1–5, IEEE,(2018).

[22] Verreist, H.; Van Acker, K.; Suykens, J.; Motmans, B.; De Moor, B.; Vandewalle, J.: NLq Neural Control Theory: Case Study for a Ball and Beam System. In: Proc. of European Control Conference (ECC), pp. 1303–1308, IEEE,(1997).

[23] Yao, S.; Liu, X.; Zhang, Y.; Cui, Z.: Research on Solving Nonlinear Problem of Ball and Beam System by Introducing Detail-Reward Function. Sym. 14, 1883(2022).

[24] Yang, J.; Wang, Q.; Li, Y.; She, J.: Stabilization of an Underactuated Ball-and-Beam System Using a Second-Order Sliding Mode Control. J. of Adv. Comp. Intel. and Intel. Info. 18, 121–127(2014).