Learning Dynamics for Control: Model-Based Reinforcement Learning of a Spring-Coupled Two-Cart Inverted Pendulum

Qinglin Yang † and Sheng Liu *,†

Posted Date: 16 March 2026

doi: 10.20944/preprints202603.1250.v1

Keywords: reinforcement learning

 doi:10.20944/preprints202603.1250.v1

Article

Learning Dynamics for Control: Model-Based Reinforcement Learning of a Spring-Coupled Two-Cart Inverted Pendulum

Qinglin Yang [1,†] and Sheng Liu [2,*,†]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
* Correspondence: sheng.liu@student.kit.edu
† These authors contributed equally to this work.

Abstract

Elastic couplings and flexible joints introduce lightly damped vibration modes that significantly complicate stabilization of nonlinear, underactuated systems. This paper studies a spring-coupled cart–inverted-pendulum benchmark inspired by the Quanser Linear Flexible Joint with Inverted Pendulum platform, where a motor-driven cart excites a passive cart through a spring–damper connection and the pendulum is mounted on the passive cart. The control objective is to stabilize the pendulum near the upright equilibrium while simultaneously regulating spring deflection and suppressing vibration. To avoid manual derivation of high-order analytical dynamics for this coupled system, we adopt a model-based reinforcement learning framework that learns task-oriented latent dynamics and performs online receding-horizon planning. Concretely, we implement Task-Oriented Latent Dynamics (TOLD) for learning a compact latent model and Temporal- Difference Model Predictive Control (TD-MPC) for MPPI-style trajectory optimization in latent space. We evaluate TD-MPC in a high-fidelity Isaac Sim / Isaac Lab simulation and compare it against a model-free PPO baseline under the same observation and action interfaces. Training curves of physical variables and returns show that TD-MPC learns coordinated balancing and spring regulation with stable convergence behavior, while PPO achieves competitive balancing performance with more pronounced non-monotonic training dynamics and transient regressions. The study highlights when online planning with learned latent models is advantageous for elastically coupled mechanisms.

Keywords: reinforcement learning

1. Introduction

Inverted pendulum systems are canonical benchmarks in control and robotics due to open-loop instability, nonlinear dynamics, and strict real-time requirements [1]. When elastic elements are introduced—e.g., flexible joints, compliant transmissions, and spring-coupled mechanisms—lightly damped vibration modes and resonance can couple with rigid-body dynamics and amplify oscillations, which is common in practical robotic linkages and mechanical transmission systems.

This work studies a spring-coupled cart–inverted-pendulum system inspired by the Quanser *Linear Flexible Joint with Inverted Pendulum* workstation. The setup comprises a motor-driven cart (driving cart), a passive cart connected via a spring–damper, and an inverted pendulum mounted on the passive cart. As the driving cart moves, spring deflection induces oscillatory behavior; the controller must simultaneously (i) stabilize the pendulum at the upright equilibrium and (ii) regulate spring deformation and vibration. Classical pipelines typically require careful force/moment derivations (e.g., Lagrangian mechanics), conversion to high-order state-space form, and parameter identification [2]; for elastically coupled systems, these steps are often cumbersome and sensitive to mismatch.

 doi:10.20944/preprints202603.1250.v1

Motivated by these limitations, we adopt a planning-centric model-based reinforcement learning (MBRL) approach that learns an implicit dynamics model from data and performs online MPC-style planning. Specifically, we implement Task-Oriented Latent Dynamics (TOLD) and Temporal-Difference Model Predictive Control (TD-MPC) [3,4]. TOLD learns a compact latent representation and latent transition model optimized for control/value estimation rather than full observation reconstruction, and TD-MPC performs receding-horizon trajectory optimization in latent space using learned reward and terminal value models.

In summary, our contributions are:

- **System formulation and objective:** We formulate control for a spring-coupled cart–pendulum benchmark with an explicit dual objective: upright stabilization and spring/vibration regulation.
- **Planning-centric MBRL implementation:** We implement TOLD+TD-MPC for this elastically coupled system, enabling online receding-horizon control without manually deriving full analytical dynamics.
- **High-fidelity comparative study:** We build an Isaac Sim/Isaac Lab articulation model and compare TD-MPC with a model-free PPO baseline under matched interfaces, reporting returns and key physical metrics during training.

2. Related Work

2.1. Classical Control of Spring-Coupled and Coupled Inverted Pendulum Systems

The inverted pendulum is a long-standing benchmark due to its instability and nonlinear dynamics. Classical methods include LQR, fuzzy control, and nonlinear control. For flexible-joint cart–pendulum systems, Xu and Choi [5] proposed an LQR-assisted fuzzy scheme that reduces the effective input dimensionality and simplifies controller design. Aggressive nonlinear strategies have also been studied. Park and Chwa [6] developed a coupled sliding-mode framework for swing-up and stabilization with semiglobal asymptotic stability guarantees under suitable conditions. For elastically coupled configurations, networked-control variants of spring-coupled inverted pendulums have served as MIMO benchmarks, where scheduling and observer-based output-feedback controllers are designed for stabilization. Remya and Jacob [7] further analyzed the influence of communication constraints on closed-loop performance. From a dynamics-and-control perspective, Semenov et al. [8] analytically characterized stability regions for two pendulums coupled by a spring and proposed a model-based stabilization method. Despite strong guarantees with accurate models, these approaches are often limited by modeling burden and mismatch in lightly damped elastic systems.

2.2. Reinforcement Learning for Inverted Pendulum and Flexible/Elastic Systems

Reinforcement learning (RL) has become an attractive alternative for pendulum control under strong nonlinearities and modeling uncertainties [9]. Model-free RL has achieved real-time swing-up and stabilization even in high-dimensional coupled setups, e.g., quadruple inverted pendulums [10]. RL has also been combined with classical control to improve robustness; Jeong and Ban [11] used RL to estimate friction parameters alongside an LQR controller. Beyond stabilization, sim-to-real RL has been used for transition control in multi-equilibrium pendulum systems under disturbances [12], and real-time experimental validations on Quanser platforms have been reported [13].

More broadly, RL is increasingly applied to flexible and compliant mechanical systems where accurate analytical modeling is difficult. Xie et al. [14] integrated a fuzzy actor–critic structure with RL for prescribed-time optimal control of flexible-joint robots, avoiding explicit model identification while handling elastic uncertainties. Experimental evidence on flexible-joint hardware has also been demonstrated, e.g., neural-network RL for rotary flexible joints [15].

2.3. Model-Based Reinforcement Learning and Learning-Based MPC

Learning-based MPC combines data-driven modeling with MPC's reliability and constraint-handling; Hewing et al. [16] provide a comprehensive overview emphasizing safety, robustness, and

 doi:10.20944/preprints202603.1250.v1

closed-loop performance. RL and MPC can be integrated by using MPC as a structured policy class; Gros and Zanon [17] proposed an actor–critic framework where the policy is parameterized by an MPC scheme and optimized via stochastic policy gradients while enforcing feasibility and constraints.

3. Preliminaries

This section summarizes the formulation and algorithmic components used in this work: the RL/MDP setup, model-based RL with latent dynamics, TD value learning, sampling-based MPC, and a model-free policy-gradient baseline.

3.1. RL and MDP Formulation

We consider continuous control as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $P(s_{t+1} \mid s_t, a_t)$ is unknown, $r(s_t, a_t)$ is the reward, and $\gamma \in (0, 1]$ is the discount factor. At time t, the agent observes s_t, samples $a_t \sim \pi(\cdot \mid s_t)$, receives reward r_t, and aims to maximize the expected discounted return

$$J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{1}$$

3.2. Model-Based RL with Latent Dynamics and TD Learning

Model-based RL (MBRL) learns a parameterized dynamics model $\hat{P}_\theta(s_{t+1} \mid s_t, a_t)$ from data to support imagined rollouts for planning and/or policy improvement, often improving sample efficiency when task-relevant dynamics are captured well. To avoid modeling directly in high-dimensional observation space, latent dynamics models learn a compact control-sufficient representation via an encoder and latent transition:

$$z_t = h_\theta(s_t), \qquad z_{t+1} = d_\theta(z_t, a_t). \tag{2}$$

Value estimation is commonly trained with temporal-difference (TD) learning. For policy π, the action-value function is

$$Q^\pi(s, a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, s_t = s, a_t = a\right], \tag{3}$$

and a one-step TD target is

$$y_t = r_t + \gamma Q(s_{t+1}, a_{t+1}), \quad a_{t+1} \sim \pi(\cdot \mid s_{t+1}). \tag{4}$$

3.3. MPC and Model-Free Baseline

Model Predictive Control (MPC) computes actions by repeatedly solving a finite-horizon optimal control problem under a predictive model, executing the first action and replanning in a receding-horizon manner. For nonlinear or learned dynamics, sampling-based MPC (e.g., MPPI) is widely used: it samples action sequences, rolls them out under the model, and reweights them according to predicted returns, which naturally handles nonlinearity and multimodal action distributions.

As a representative model-free baseline, we use Proximal Policy Optimization (PPO), an on-policy policy-gradient method that stabilizes updates via a clipped surrogate objective.

4. Method

4.1. Task Setup

We formulate spring-coupled cart–pendulum control as a continuous-control MDP. At each step t, the agent observes $s_t \in \mathbb{R}^9$ and outputs a scalar action $a_t \in \mathbb{R}$ that commands the effort at the actuated prismatic joint of the driving cart. With $\gamma = 0.99$, the goal is to maximize $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ while (i) stabilizing the pendulum near the upright equilibrium and (ii) regulating spring deflection and vibration.

 doi:10.20944/preprints202603.1250.v1

4.2. Reward

We encode upright stabilization and elastic regulation by penalizing spring deflection $(L_t - L_0)$, spring-velocity vibration proxy $\dot{L}_t$, and control effort while encouraging uprightness:

$$r_t = w_\theta \rho(\theta_t) - w_L(L_t - L_0)^2 - w_{\dot{L}} \dot{L}_t^2 - w_u a_t^2, \tag{5}$$

where $\rho(\theta_t)$ is a bounded uprightness shaping term and weights are shared across methods.

4.3. Planning-Centric MBRL: TOLD + TD-MPC

Our primary approach combines Task-Oriented Latent Dynamics (TOLD) with Temporal-Difference Model Predictive Control (TD-MPC) [3,4]. TOLD learns a task-relevant latent model and TD-MPC performs MPPI-style receding-horizon planning in latent space.

TOLD uses an encoder, latent transition, reward model, terminal action-value, and a policy used for guided rollouts:

$$z_t = h_\theta(s_t), \qquad z_{t+1} = d_\theta(z_t, a_t), \tag{6}$$

$$\hat{r}_t = R_\theta(z_t, a_t), \qquad \hat{q}_t = Q_\theta(z_t, a_t), \qquad \hat{a}_t \sim \pi_\theta(z_t). \tag{7}$$

We use an encoder dimension of 256, an MLP hidden dimension of 512, and a latent dimension of 50. Training is off-policy with replay size 100,000 and batch size 512. For horizon $H = 5$, we encode once and unroll in latent space, matching inference-time usage. We optimize a temporally weighted objective (weight $\rho = 0.5$):

$$\mathcal{J}(\theta; \Gamma) = \sum_{i=t}^{t+H} \rho^{\,i-t}\, \mathcal{L}(\theta; \Gamma_i), \tag{8}$$

with per-step loss

$$\begin{aligned} \mathcal{L}(\theta; \Gamma_i) = \alpha_r \| R_\theta(z_i, a_i) - r_i \|_2^2 \\ + \alpha_v \Big\| Q_\theta(z_i, a_i) - \\ \Big(r_i + \gamma Q_{\theta^-}\big(z_{i+1}, \pi_\theta(z_{i+1})\big) \Big) \Big\|_2^2 \\ + \alpha_c \| d_\theta(z_i, a_i) - h_{\theta^-}(s_{i+1}) \|_2^2, \end{aligned} \tag{9}$$

where $(\alpha_r, \alpha_v, \alpha_c) = (0.5, 0.1, 2.0)$, $\gamma = 0.99$, gradient clipping norm 10, and target update rate $\tau = 0.01$; prioritized replay uses $(\alpha, \beta) = (0.6, 0.4)$.

At each control step, TD-MPC optimizes action sequences in latent space with planning horizon $H = 5$, iterations $J = 6$, $N = 512$ samples, and $k = 64$ elites, injecting policy-guided rollouts with mixture coefficient 0.05. Candidate trajectories are scored by

$$\phi_\Gamma = \sum_{h=0}^{H-1} \gamma^h R_\theta(z_{t+h}, a_{t+h}) + \gamma^H Q_\theta(z_{t+H}, a_{t+H}), \tag{10}$$

with elite weights $\Omega_i = \exp(\eta \phi_{\Gamma_i^\star})$ and $\eta = 0.5$. We enforce a minimum action-sequence standard deviation of 0.05, use a linear std schedule $\mathrm{linear}(0.5, 0.05, 25{,}000)$ and a linear horizon schedule $\mathrm{linear}(1, 5, 25{,}000)$, and set momentum to 0.1.

5. Experiments

This section reports simulation results on the spring-coupled cart–inverted-pendulum system, inspired by the Quanser *Linear Flexible Joint with Inverted Pendulum* workstation (Figure 1), which combines a flexible mass–spring subsystem with an inverted-pendulum benchmark. All experiments are conducted in the same Isaac Sim/Isaac Lab environment with matched initial-condition distributions for fair comparison.

 doi:10.20944/preprints202603.1250.v1

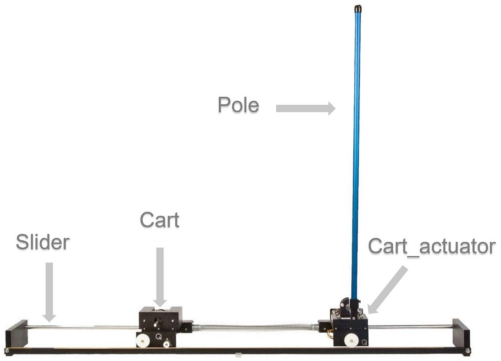

Figure 1. Linear Flexible Joint with Inverted Pendulum.

We study stabilization around the upright equilibrium, initializing each episode near upright with small random perturbations in pendulum state, cart positions, and spring deformation. We compare a planning-centric model-based controller (TD-MPC) against a model-free baseline (PPO) under identical observation/action interfaces, reward, and termination conditions, and evaluate performance via training curves of key physical variables and returns.

5.1. Model Building in Isaac Sim

We construct the spring-coupled cart–inverted-pendulum model in NVIDIA Isaac Sim and perform training/evaluation in Isaac Lab (Figure 2). The system is authored in a single USD stage and represented as an open-chain PhysX articulation rooted at `/World/cartpole` (marked as a `Physics Articulation Root`). A fixed rail prim (`slider`) anchors the articulation to the world via a `PhysicsFixedJoint`.

Two rigid carts, `cart_actuator` (actuated) and `cart` (passive), are modeled as box rigid bodies with configured mass/inertia and collision/visual geometries. The actuated cart is connected to the rail through a prismatic joint aligned with the track direction; this is the only directly actuated joint, consistent with PhysX open-chain constraints. The elastic coupling between the carts is implemented as a prismatic joint (`cart_to_cart1`) equipped with a linear drive whose `stiffness`, `damping`, and `target position` parameters specify the spring–damper behavior and rest length.

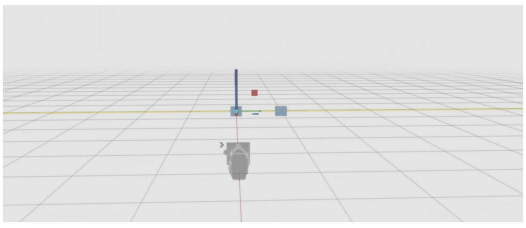

Figure 2. USD model.

The pendulum link `pole` is modeled as a slender rigid body and attached to the passive cart via a revolute joint (`cart_to_pole`) with its axis perpendicular to the cart motion. Small joint stiffness, damping, and friction are added to emulate hinge dissipation and avoid unrealistically undamped oscillations. This modeling procedure yields a high-fidelity open-chain articulation that preserves the key coupled rigid-body and elastic dynamics of the Quanser *Linear Flexible Joint with Inverted Pendulum* setup.

5.2. Experimental Setup

We evaluate TD-MPC and PPO in the same environment, using identical observation/action interfaces and an identical reward structure. For TD-MPC, we train for 100,000 environment steps with episode length 500 and action repeat 4, using random seed 1 and a single environment instance. For PPO, we use the same environment under identical observation/action interfaces and the same reward structure, with random seed 1, and train up to 16,667 iterations (24 steps per environment per iteration).

 doi:10.20944/preprints202603.1250.v1

5.3. Evaluation Metrics

We track both task-level performance and physically meaningful signals:

- **Balancing quality:** pendulum angle (deg) and angular velocity (rad/s).
- **Elastic regulation:** spring length, spring deformation $(L_t - L_0)$, and spring velocity $\dot{L}_t$.
- **Learning behavior:** running mean reward and episodic return during training.

These quantities directly reflect the dual objective of upright stabilization and vibration suppression.

5.4. TD-MPC Results

We record key physical variables and reward signals throughout training. Figure 3 shows the evolution of spring deformation/length and pendulum angle/angular velocity, and Figure 5 reports reward curves. In early training, exploration in this coupled, lightly damped nonlinear system induces large oscillations in both the spring and pendulum states. As training progresses, TD-MPC learns coordinated regulation: spring deformation decreases toward zero (spring length stabilizes near L_0) and the pendulum converges toward the upright equilibrium with reduced angular velocity. A clear convergent trend is observed after approximately 2.3×10^4 steps, characterized by small steady-state fluctuations and increased cumulative reward. Overall, the results indicate that the reward design and planning-centric control successfully yield stable, physically consistent behavior in the elastically coupled system.

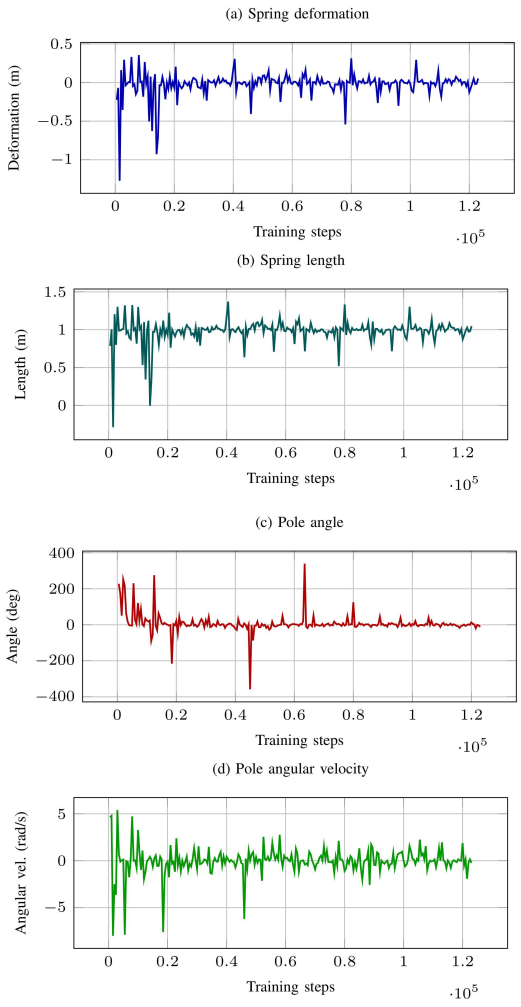

Figure 3. Training curves of key physical state variables for TD-MPC. (a) Spring deformation; (b) spring length; (c) pole angle; (d) pole angular velocity.

 doi:10.20944/preprints202603.1250.v1

5.5. PPO Results

As shown in Figure 4 and Figure 6, the learning curve is non-monotonic but improves on average, which is typical for on-policy methods due to sampling noise and distribution shift after each update. PPO reaches a relatively stable balancing regime after about 9,000 steps, with increasing mean return and the pendulum angle approaching 0°. We observe a transient degradation between roughly 15,000 and 17,000 steps, where returns drop and angle oscillations increase, followed by recovery. This behavior is consistent with sensitivity to update intensity when the policy becomes more deterministic (e.g., multiple epochs on the same on-policy batch and clipping effects) [18]. The entropy-loss curve is therefore treated as an optimization diagnostic rather than a direct performance metric.

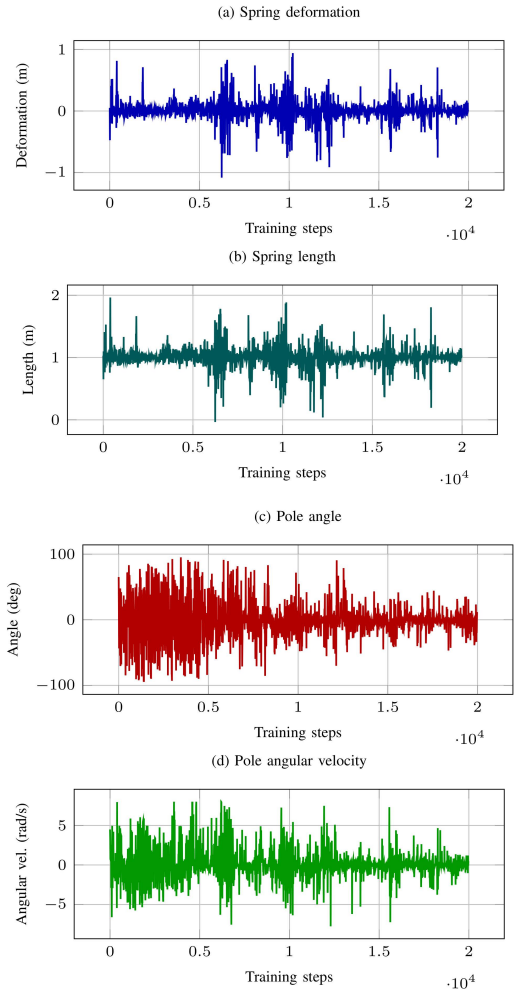

Figure 4. Training curves of key physical state variables for PPO. (a) Spring deformation; (b) spring length; (c) pole angle; (d) pole angular velocity.

 doi:10.20944/preprints202603.1250.v1

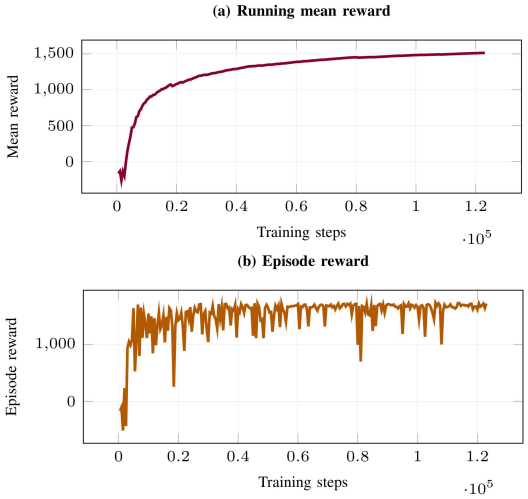

Figure 5. Evolution of reward signals during TD-MPC training. (a) Running mean reward; (b) episode reward.

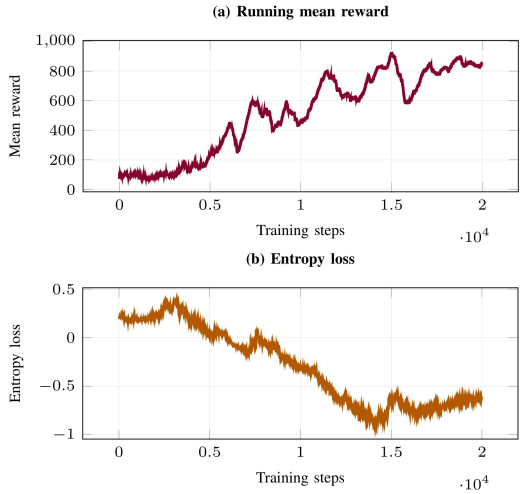

Figure 6. Reward and entropy loss curves during PPO training. (a) Running mean reward; (b) entropy loss.

5.6. Comparative Analysis: TD-MPC vs. PPO

TD-MPC and PPO represent two distinct paradigms: TD-MPC is model-based, off-policy, and planning-centric, whereas PPO is model-free and on-policy [19,20]. TD-MPC improves data efficiency by reusing experience from a replay buffer and by learning latent dynamics/reward/value models, while receding-horizon planning helps limit error accumulation and mitigate model bias. In contrast, PPO avoids model bias by optimizing directly on environment rollouts, but discards data after each update and can require more interaction and careful tuning to reach comparable performance. At inference time, the trade-off reverses: TD-MPC incurs higher latency due to sampling-based online planning, whereas PPO requires only a single policy forward pass and is thus easier to deploy at high control rates. Regarding constraints and safety, TD-MPC can incorporate constraints more explicitly within planning (e.g., via penalties or restricted sampling), while PPO typically enforces constraints indirectly through reward shaping and termination conditions.

In our experiments, TD-MPC reaches the predefined performance threshold in approximately 24 minutes, whereas PPO requires about 90 minutes to achieve comparable performance, indicating substantially better wall-clock sample efficiency for TD-MPC (Table 1). Table 2 summarizes final performance: TD-MPC converges to a higher mean reward and maintains it, while PPO exhibits a larger gap between peak and final performance.

 doi:10.20944/preprints202603.1250.v1

Table 1. Training efficiency comparison between PPO and TD-MPC.

Algorithm	Time to Threshold (min)	Env Steps to Threshold
PPO (stabilized)	90	10600
TD-MPC	**24**	**24000**

Table 2. Final performance comparison between PPO and TD-MPC.

Algorithm	Best Mean Reward	Final Mean Reward
PPO (stabilized)	921	853
TD-MPC	**1541**	**1541**

Training stability metrics in Table 3 further confirm this trend: PPO shows larger Peak–Final Drop and Worst Post-Peak Dip, whereas TD-MPC exhibits minimal post-peak degradation. Finally, Table 4 reports qualitative observations of policy instability.

Table 3. Quantitative comparison of training stability and reward degradation.

Algorithm	Peak–Final Drop	Worst Post-Peak Dip
PPO (stabilized)	56	304
TD-MPC	**4**	**13**

Table 4. Qualitative observation of policy instability during training.

Algorithm	Collapse Observed
PPO (stabilized)	Mild
TD-MPC	No

6. Conclusions and Future Work

This paper studied learning-based control of an elastically coupled cart–inverted-pendulum system inspired by the Quanser *Linear Flexible Joint with Inverted Pendulum* platform, where a motor-driven cart excites a passive cart through a spring–damper connection and the pendulum is mounted on the passive cart. The controller must simultaneously stabilize the pendulum near the upright equilibrium and suppress spring deflection/vibration. To reduce reliance on cumbersome analytical modeling, we adopted a planning-centric model-based RL approach combining TOLD with TD-MPC and compared it to a model-free PPO baseline under matched interfaces and a shared reward. In high-fidelity Isaac Sim/Isaac Lab experiments, TD-MPC learned coordinated regulation that stabilizes the pendulum while driving the spring toward its equilibrium and attenuating oscillations, with more consistent convergence in both physical signals and returns than PPO. PPO achieved competitive balancing with low inference latency but exhibited more variable, non-monotonic training dynamics. Future work will focus on rigorous multi-seed quantitative evaluation under systematic parameter variations and disturbances, improving TD-MPC real-time feasibility and robustness to model bias, and validating the approach on hardware via sim-to-real techniques (e.g., domain randomization and online adaptation) and more explicit constraint/safety integration.

Conflicts of Interest: 'The authors declare no conflicts of interest.

References

1. Boubaker, O. International Journal of Advanced Robotic Systems the Inverted Pendulum Benchmark in Nonlinear Control Theory: a Survey Regular Paper.

 doi:10.20944/preprints202603.1250.v1

2. jun Wang, J. Simulation studies of inverted pendulum based on PID controllers. *Simul. Model. Pract. Theory* **2011**, *19*, 440–449.
3. Hansen, N.; Wang, X.; Su, H. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955* **2022**.
4. Hansen, N.; Su, H.; Wang, X. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828* **2023**.
5. Xu, Y.; Choi, B.J. Control of Flexible Joint Cart based Inverted Pendulum using LQR and Fuzzy Logic System. *Journal of the Korean Institute of Intelligent Systems* **2013**, *23*, 268–274.
6. Park, M.S.; Chwa, D. Swing-up and stabilization control of inverted-pendulum systems via coupled sliding-mode control method. *IEEE transactions on industrial electronics* **2009**, *56*, 3541–3555.
7. Remya, P.; Jacob, J. Static network access scheduling and stabilization of two inverted pendulums spring coupled networked control system. In Proceedings of the 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies. IEEE, 2014, pp. 98–103.
8. Semenov, M.E.; Solovyov, A.M.; Popov, M.A.; Meleshenko, P.A. Coupled inverted pendulums: stabilization problem. *Archive of Applied Mechanics* **2018**, *88*, 517–524.
9. Kiumarsi, B.; Vamvoudakis, K.G.; Modares, H.; Lewis, F.L. Optimal and Autonomous Control Using Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* **2018**, *29*, 2042–2062. https://doi.org/10.1109/TNNLS.2017.2773458.
10. Oh, Y.; Lee, T.; Ryoo, S.; Koh, K.C.; Han, S.; Lee, Y.S. Reinforcement Learning to Achieve Real-time Control of a Quadruple Inverted Pendulum. *International Journal of Control, Automation and Systems* **2025**, *23*, 2797–2806.
11. Jeong, J.; Ban, J. Reinforcement learning-based friction compensation of an inverted pendulum on a cart. *International Journal of Machine Learning and Cybernetics* **2025**, pp. 1–19.
12. Lee, T.; Ju, D.; Lee, Y.S. Transition control of a double-inverted pendulum system using Sim2Real reinforcement learning. *Machines* **2025**, *13*, 186.
13. BAJRAMI, X.; KAÇIU, F.; SHALA, E.; LIKAJ, R. Real-Time Swing-up of a Linear Inverted Pendulum Using Reinforcement Learning. *Mechanics* **2025**, *31*, 123–135.
14. Xie, S.; Sun, W.; Sun, Y.; Su, S.F. Adaptive Prescribed-Time Optimal Control for Flexible-Joint Robots via Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2025**.
15. Sendrescu, D.; Bujgoi, G.; Chintescu, D. Control of a rotary flexible joint experiment based on reinforcement learning. In Proceedings of the 2020 21th International Carpathian Control Conference (ICCC). IEEE, 2020, pp. 1–5.
16. Hewing, L.; Wabersich, K.P.; Menner, M.; Zeilinger, M.N. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems* **2020**, *3*, 269–296.
17. Gros, S.; Zanon, M. Reinforcement learning based on MPC and the stochastic policy gradient method. In Proceedings of the 2021 American Control Conference (ACC). IEEE, 2021, pp. 1947–1952.
18. Moalla, S.; Miele, A.; Pyatko, D.; Pascanu, R.; Gulcehre, C. No representation, no trust: connecting representation, collapse, and trust issues in ppo. *Advances in Neural Information Processing Systems* **2024**, *37*, 69652–69699.
19. Swazinna, P.; Udluft, S.; Hein, D.; Runkler, T. Comparing model-free and model-based algorithms for offline reinforcement learning. *IFAC-PapersOnLine* **2022**, *55*, 19–26.
20. Huys, Q.J.; Seriès, P. Reward-based learning, model-based and model-free. In *Encyclopedia of Computational Neuroscience*; Springer, 2022; pp. 3042–3050.