arXiv:2611.15133v1 [eess.SY] 11 Nov 2026

Sim-to-Real Powertrain Vibration Control via Model-Based Assisted Domain-Randomized Reinforcement Learning

Heisei Yonezawa, *Member, IEEE,* Ansei Yonezawa, *Member, IEEE,* Itsuro Kajiwara

***Abstract*—Controllers designed from nominal models often degrade on mechanical systems with nonlinearities and parametric variations, such as automotive powertrains with backlash. Deep reinforcement learning (DRL) with domain randomization can acquire robustness to such model discrepancies, but wide randomization destabilizes training and yields conservative policies, and experimental transfer of such policies has rarely been demonstrated in vibration control. This paper presents model-based controller assisted domain randomization (MBCADR), a framework that integrates model-based control into domain-randomized DRL. An H_2 controller designed from the nominal model supplies a baseline action, while a residual policy trained with twin delayed deep deterministic policy gradient under randomized dynamics compensates for nonlinearities and uncertainties. This model-based assistance relieves the agent of synthesizing the entire control input, thereby stabilizing training under extensive randomization. For each realization of the uncertain parameters, the augmented plant–controller state is shown to be a Markov state of the closed loop, so that domain-randomized training is rigorously formulated as policy optimization over a parameterized family of Markov decision processes. A Lyapunov-based stability analysis further derives a forward-invariant set and an explicit ultimate bound, in which the learned action enters only through its saturation level. Monte Carlo simulations with multiple training seeds show that MBCADR reduces performance variability compared with either component alone. Experiments on a powertrain test bench with backlash, using the simulation-trained policy on an embedded processor without further tuning, confirm successful sim-to-real transfer.**

***Index Terms*—Active vibration control, deep reinforcement learning, domain randomization, model-based control, powertrain, sim-to-real transfer.**

I. Introduction

A. Background and Motivation

INDUSTRIAL mechanical systems, such as automotive powertrains [1], [2] and robotic manipulators [3], [4], are subject to increasingly stringent performance requirements, and their dynamics have become correspondingly complex. Nonlinearities, time delays, and variations in physical parameters are inherent in such systems [5], [6], and controllers must ensure stability and performance in their presence. A representative example is the automotive powertrain, in which the torque generated by the actuator is transmitted to the wheels through compliant shafts and gears, so that abrupt torque transients excite torsional vibration that degrades drivability. Gear backlash further aggravates this problem by inducing switching between contact and non-contact modes, which renders the dynamics discontinuous [7], [8].

Since the advent of state-space theory [9], model-based control has been the mainstay of industrial practice, and its performance is fundamentally governed by model fidelity. Robust control methods such as H_∞ control [10]–[13] guarantee performance under uncertainty, but only for uncertainties that can be represented in the form required by the synthesis. Discontinuous nonlinearities such as backlash, and simultaneous variations in multiple parameters, are difficult to capture in this form without conservative designs. Moreover, modeling and calibration errors can never be fully eliminated, so a controller that performs well in simulation may degrade when deployed on the physical plant. This problem is known as the sim-to-real gap [14].

Deep reinforcement learning (DRL) offers an alternative that does not require an explicit transition model, since the agent improves its policy from data collected through interaction with the environment [15]–[17]. DRL has been applied to vibration suppression of flexible link manipulators [18], flexible rotors [19], magnetorheological elastomer absorbers [20], flexible hinged plates [21], and robot control [22]. It has also been combined with conventional controllers, for example with a proportional–integral–derivative (PID) controller [23] and with active disturbance rejection control [24].

Training a DRL agent directly on hardware is rarely practical: the exploratory actions required for learning may drive the plant into unsafe operating regions, and the required volume of interaction data is costly to acquire [14], [25]. Agents are therefore commonly trained in simulation, where exploration is safe and data are virtually unlimited. A policy trained in this manner, however, is just as susceptible to the sim-to-real gap as a model-based controller.

B. Related Work

Domain randomization (DR) addresses the sim-to-real gap by randomizing the physical parameters of the simulator at the start of each training episode [14], [26]. The policy is then optimized over a set of Markov decision processes (MDPs) rather than a single one, which promotes robustness to parameter variations within the randomized range. A theoretical analysis has also clarified the conditions under which DR achieves successful transfer [27]. DR has been applied successfully to object pushing with a 7-DOF manipulator [14], deformable object manipulation [28], bipedal locomotion [29], [30], legged locomotion learned from animal motion [31], humanoid robots [32], [33], unmanned aerial vehicle [34], path following of underwater vehicles [35], and overhead crane control under payload variations [26].

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI [Grant Number 23K13273].

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Previous DR studies leave two issues from a control-engineering standpoint. First, DR has been applied predominantly in robotics, whereas its use for vibration control of mechanical systems remains scarce. Systems with discontinuous nonlinearities, such as powertrains with backlash, have been addressed only in a few studies, including the authors' previous work [36]. Second, because the training environment changes from episode to episode, wide randomization makes the learning problem harder [28], destabilizes training, and tends to produce overly conservative policies [37]–[39]. Enlarging the network and the amount of training data can offset these effects in principle, but at the cost of longer development, a higher risk of overfitting, and policies that are too large for the memory-constrained embedded processors on which automotive control laws are executed.

Combining a model-based controller with a learning component is a promising way to address these issues. This idea was formalized as residual reinforcement learning by Johannink *et al.* [40] and Silver *et al.* [41], in which a policy learns a corrective action that is added to the output of an existing controller. Johannink *et al.* showed on a real robot that this structure accomplishes a block assembly task involving contacts that are difficult to model. Silver *et al.* showed that residual learning substantially improves imperfect controllers in complex manipulation tasks for which learning from scratch is data-inefficient. Koryakovskiy *et al.* [42] used reinforcement learning to compensate for the model–plant mismatch of a nonlinear model predictive controller and validated the approach on a real robot. Other combinations of model-based control and reinforcement learning have also been reported [43], [44]. Cao *et al.* [45] combined linear state feedback with a residual policy trained using a Lyapunov-like reward, and derived conditions under which a safety envelope is rendered invariant and the closed loop is stabilized. Although some of these studies introduce variations in task conditions, a formulation as DR that substantially varies the dynamics of the controlled plant itself has not been investigated in depth. Moreover, the application of these hybrid schemes to automotive systems and vibration control remains largely unexplored.

Liu *et al.* [46] combined a PID controller with a residual policy for autonomous blimp control, randomized wind and buoyancy during training, and transferred the simulation-trained policy to a real blimp. Zuo *et al.* [47] replaced the PID controller with an H_∞ controller to enlarge the stability margin, and derived a lower bound on the mixing factor between the two actions that ensures stability under assumptions on the distribution of the learned actions. Kim *et al.* [48] combined nonlinear model predictive control and a disturbance observer with a residual policy trained under randomized payloads and friction, and bounded the learned action on the basis of an input-to-state stability analysis. Yan *et al.* [49] augmented a model-based bipedal locomotion controller with a residual policy trained under dynamics randomization, and accelerated learning through a supervised loss derived from a model-based oracle with privileged access to the true dynamics. However, these hybrid schemes are formulated for a particular plant and a particular baseline controller, often at the cost of a more complex control structure with additional components such as disturbance observers, estimator networks, and privileged information; a general formulation of the domain-randomized optimization problem for the resulting closed loop has not been established. Moreover, the stability of the closed loop including the learned policy has been addressed only in limited settings.

In the authors' previous work, Yonezawa *et al.* [36] proposed a hybrid approach combining domain-randomized DRL and output-feedback H_2 vibration control and showed in simulation that it achieves robustness with a compact network and limited training data. A preliminary version of this work [50] introduced TD3 into the framework. Both studies, however, were confined to numerical simulation, and two questions remained open. First, it was not shown whether adding a learned action to a model-based closed loop preserves the stability that the model-based controller provides. Second, it was not shown whether the robustness obtained through DR carries over to physical hardware when the policy runs on an embedded processor.

C. Contributions of This Study

This paper proposes model-based controller assisted domain randomization (MBCADR), a robust control framework that integrates model-based control into domain-randomized DRL, as outlined in Fig. 1. DRL does not replace the model-based controller. Instead, it learns corrective actions for modeling errors, while the model-based controller provides the baseline action from the start of training. With this division of roles, the agent only needs to learn compensation for model discrepancies rather than the entire control input. This reduction of the learning burden becomes increasingly important as the randomization range widens, because wider randomization makes training harder. Although an automotive powertrain is used as the application example, the framework is formulated for general nonlinear systems with state-space model-based controllers. This paper extends the preliminary version [50] with a closed-loop stability analysis, Monte Carlo evaluation over multiple training seeds, and experimental validation on a physical test bench.

The main contributions are as follows:

1) *Framework and formulation:* A simple control framework that strengthens DR through the assistance of a model-based controller designed from a nominal model. For each realization of the uncertain parameters, the augmented plant–controller state is shown to satisfy the Markov property. Domain-randomized training is therefore formulated as policy optimization over a family of MDPs parameterized by the uncertain physical parameters.
2) *Stability analysis:* A Lyapunov-based analysis of the hybrid closed loop derives a forward-invariant set and an explicit ultimate bound. These are computed in advance from the design setting, without modifying the learned policy online. The learned action enters the bound only through its saturation level, so the trade-off between learning authority and the guaranteed bound can be evaluated in advance for any internally stabilizing model-based controller.

3) *Experimental sim-to-real validation:* Monte Carlo simulations with multiple training seeds show that MBCADR reduces performance variability compared with model-based control alone and with DR alone. Experiments on a powertrain test bench with backlash under several hardware configurations confirm successful sim-to-real transfer. The simulation-trained policy, a compact network that fits in the memory of an embedded processor, runs on the processor without further tuning.

The remainder of this paper is organized as follows. Section II formulates the control problem. Section III reviews the MDP and TD3. Section IV presents MBCADR. Section V analyzes the stability of the hybrid closed loop. Sections VI and VII report the simulation and experimental results, respectively. Section VIII concludes the paper.

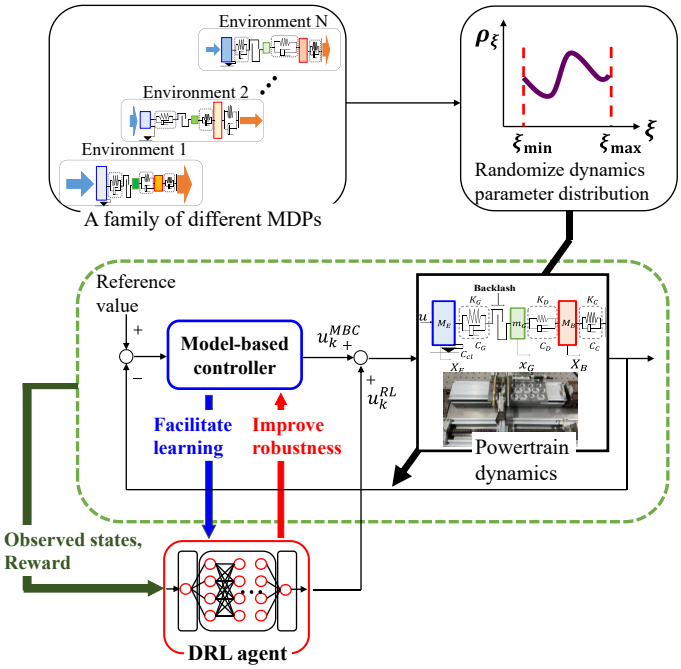


Fig. 1. Outline of the proposed approach.

II. Problem Formulation

Consider a discrete-time nonlinear system whose dynamics depend on uncertain physical parameters:

$$x_{k+1} = f(x_k, u_k, w_k; \xi) \tag{1}$$

$$y_k = h(x_k; \xi) \tag{2}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $w_k \in \mathbb{R}^l$, and $y_k \in \mathbb{R}^o$ denote the state, control input, exogenous disturbance, and controlled output at time step k, respectively. The maps $f\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^n$ and $h\colon \mathbb{R}^n \to \mathbb{R}^o$ describe the state transition and the output, and neither is known exactly to the designer. The vector ξ collects the uncertain physical parameters of the plant, such as inertial and damping parameters and the width of a dead zone, which also govern the nonlinear effects. In the simulator used for training, the dynamics are parameterized by the same vector ξ, so that each training episode is specified by a realization of ξ (see Section IV-B).

The control objective is output tracking: for a given reference $y_k^r \in \mathbb{R}^o$, the tracking error $e_k := y_k^r - y_k$ should converge to zero, that is, $\lim_{k\to\infty} \|e_k\| = 0$. This objective is cast as the finite-horizon optimal control problem

$$\min_{\{u_0,u_1,\ldots,u_T\}\in U} \quad \sum_{k=0}^{T} c(x_k, u_k) \qquad \text{subject to} \quad (1),(2),\ k = 0, 1, \ldots, T \tag{3}$$

where U is the admissible input set. The stage cost $c\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is nonnegative, vanishes when $e_k = 0$ and $u_k = 0$, and grows unboundedly as $\|(e_k, u_k)\| \to \infty$. In this study, the quadratic cost

$$c(x_k, u_k) = e_k^\top Q e_k + u_k^\top R u_k \tag{4}$$

is adopted, where $Q \in \mathbb{R}^{o\times o}$ is positive semidefinite and $R \in \mathbb{R}^{m\times m}$ is positive definite.

Because f and h are not known exactly, (3) cannot be solved directly by conventional model-based design. In many mechanical systems, however, partial knowledge of the dynamics is available. The following assumptions formalize this knowledge.

Assumption 1 (Nominal model). A linear nominal model of (1) and (2) is available:

$$x_{k+1} = Ax_k + B_1 w_k + B_2 u_k \tag{5}$$

$$y_k = Cx_k \tag{6}$$

where $A \in \mathbb{R}^{n\times n}$, $B_1 \in \mathbb{R}^{n\times l}$, $B_2 \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{o\times n}$. Such a model can be obtained from first-principles modeling or system identification at nominal parameter values.

Assumption 2 (Model-based controller). A discrete-time linear controller of the form

$$x_{k+1}^c = A_c x_k^c + B_c e_k \tag{7}$$

$$u_k^{MBC} = C_c x_k^c + D_c e_k \tag{8}$$

with internal state $x_k^c \in \mathbb{R}^{n_c}$ and matrices $A_c \in \mathbb{R}^{n_c\times n_c}$, $B_c \in \mathbb{R}^{n_c\times o}$, $C_c \in \mathbb{R}^{m\times n_c}$, and $D_c \in \mathbb{R}^{m\times o}$, is available such that the feedback interconnection of (5)–(8) is internally stable.

Assumption 2 specifies only the structure of the controller; any synthesis method that yields an internally stabilizing controller in the form of (7) and (8) may be used. The specific controller adopted in this study is described in Section IV-A.

The discrepancy between the true dynamics (1) and the nominal model (5) is denoted by

$$\Delta_k := f(x_k, u_k, w_k; \xi) - (Ax_k + B_1 w_k + B_2 u_k). \tag{9}$$

This term collects the effects neglected in (5), namely nonlinearities such as backlash and the deviations of ξ from the nominal values. Because the controller (7) and (8) is designed without knowledge of Δ_k, its tracking performance can degrade, and stability can even be lost, when it is applied to the true plant (1) and (2). The objective of this paper is to retain the model-based controller (7) and (8) while compensating for Δ_k through learning.

III. Reinforcement Learning Preliminaries

A. Markov Decision Process

Reinforcement learning problems are formulated as a Markov decision process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $p(s_{k+1} \mid s_k, a_k)$ the state transition probability, $r(s_k, a_k)$ the reward function, and $0 < \gamma < 1$ the discount factor [15], [51], [52]. At each discrete time step k, the agent observes a state s_k, selects an action a_k according to a policy π, and receives a reward r_k. The objective is to maximize the expected cumulative discounted reward

$$J(\pi) = \mathbb{E}\left[\sum_{i=1}^{T} \gamma^{i-1} r(s_i, a_i) \mid s_1 = s, a_1 = a; \pi\right] \tag{10}$$

In this study, the reward is defined from the stage cost (4) with the opposite sign, so that maximizing (10) corresponds to minimizing the cost in (3).

B. Twin Delayed Deep Deterministic Policy Gradient

For continuous-action control problems, a deterministic policy can be learned by actor–critic methods using off-policy data [51], where a critic estimates the value function. In this study, Twin Delayed Deep Deterministic Policy Gradient (TD3) [53] is employed as the learning algorithm for the reinforcement learning component. TD3 uses a deterministic policy (*actor*) $\mu_\theta(\cdot)$ and two action-value functions (*critics*) $Q_{\phi_1}(\cdot)$ and $Q_{\phi_2}(\cdot)$. The actor outputs a continuous action $a_k = \mu_\theta(s_k)$. To achieve sample-efficient learning, TD3 is trained off-policy using transitions stored in a replay buffer $\mathcal{D}$. To reduce overestimation bias, TD3 maintains two critics and uses a conservative target based on their minimum. Given a sampled transition $(s_k, a_k, r_k, s_{k+1}) \sim \mathcal{D}$, the target action is computed using a slowly-updated target actor μ_{θ^-} with a smoothed perturbation:

$$\begin{aligned} \tilde{a}_{k+1} &= \mu_{\theta^-}(s_{k+1}) + \varepsilon \\ \varepsilon &\sim \mathrm{clip}\big(\mathcal{N}\big(0, \sigma^2\big), -c, c\big). \end{aligned} \tag{11}$$

Then, the target value is defined as

$$Y_k = r_k + \gamma \min_{i \in \{1,2\}} Q_{\phi_i^-}(s_{k+1}, \tilde{a}_{k+1}) \tag{12}$$

where $Q_{\phi_i^-}$ denotes target critics. Each critic is updated by minimizing the temporal-difference loss

$$L(\phi_i) = \mathbb{E}_{(s_k, a_k, r_k, s_{k+1}) \sim \mathcal{D}}\Big[\big(Q_{\phi_i}(s_k, a_k) - Y_k\big)^2\Big], \quad i \in \{1, 2\}. \tag{13}$$

The actor is updated less frequently than the critics, i.e., delayed policy updates, to stabilize the critic output. Using the chain rule, the deterministic policy gradient is computed as

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s_k \sim \mathcal{D}}\big[\nabla_\theta \mu_\theta(s_k) \nabla_a Q_{\phi_{min}} \mid_{a = \mu_\theta(s_k)}\big] \tag{14}$$

where $Q_{\phi_{min}}$ denotes the critic that attains $\min(Q_{\phi_1}, Q_{\phi_2})$ at $(s_k, \mu_\theta(s_k))$.

IV. Proposed Approach

A. Model-Based Controller and Hybrid Control Input

In this study, the controller in Assumption 2 is an H_2 controller synthesized from the nominal model by the LMI approach [10], [54]. Any other controller satisfying Assumption 2 can be used without changing the subsequent formulation. To compensate for the model mismatch Δ_k, a learning-based input is added to the model-based input, and the overall control input is defined as

$$u_k = u_k^{MBC} + u_k^{RL} \tag{15}$$

where u_k^{RL} is generated by a reinforcement learning policy.

The two components in (15) play complementary roles. From the viewpoint of the learning agent, the MBC supplies a stabilizing action derived from the nominal model from the first training episode. Because the closed loop already achieves reasonable tracking before learning starts, the agent avoids a large number of uninformative trials that an immature policy would otherwise generate, and exploration is concentrated around a meaningful operating region. This assistance is particularly valuable under DR, in which the environment changes from episode to episode and learning the entire control law from scratch becomes difficult [28]. From the viewpoint of the MBC, the learning agent compensates for the model mismatch Δ_k that the MBC cannot address because it is designed only from the nominal model.

This division of roles reduces the burden on the learning agent: the agent focuses on compensating for residual dynamics rather than synthesizing the entire control input, so that the function to be approximated by the policy becomes simpler. As a result, a compact network and a limited amount of training data suffice [36]. Moreover, because the nominal performance is secured by the MBC, the policy is not required to achieve robustness at the expense of the baseline performance, which is expected to mitigate the conservativeness that DR tends to induce [37]–[39]. Finally, the learned correction u_k^{RL} is bounded by the saturating output layer of the actor, so that it acts only within a prescribed range around the baseline action; this property is exploited in the stability analysis of Section V.

B. Domain Randomization and Augmented Dynamics

During training, domain randomization is employed to address parametric uncertainty. At the beginning of each episode, the parameter vector ξ is randomly sampled from a predefined distribution ρ_ξ and remains fixed throughout the episode [14], [26].

Standard reinforcement learning algorithms, including TD3, are derived under the premise that the environment is an MDP, that is, the next state depends only on the current state and action [15]. It is therefore desirable that the training environment satisfies this property [43]. Under the hybrid input (15), however, the agent does not interact with the plant alone: the MBC, which has its own internal state, is inserted between the agent and the plant. It must therefore be confirmed that the environment seen by the agent remains an MDP.

From the viewpoint of the agent, the nonlinear plant and the MBC together constitute a single environment, as illustrated in

Fig. 1. The agent supplies u_k^{RL} to this closed-loop system and receives signals generated within it. Consequently, not only the plant output but also internal signals of the MBC can be included in the observation of the agent. The state of this environment is obtained by augmenting the plant state with the controller state:

$$\mathcal{X}_k^{env} := \begin{bmatrix} x_k^\top & x_k^{c\top} \end{bmatrix}^\top, \tag{16}$$

and the closed-loop dynamics under the hybrid control input [36] are written as

$$\begin{aligned} \mathcal{X}_{k+1}^{env} &= \begin{bmatrix} x_{k+1} \\ x_{k+1}^c \end{bmatrix} = \begin{bmatrix} f(x_k, u_k, w_k; \xi) \\ A_c x_k^c + B_c e_k \end{bmatrix} \\ &= \begin{bmatrix} f\big(x_k, u_k^{RL} + u_k^{MBC}, w_k; \xi\big) \\ A_c x_k^c + B_c e_k \end{bmatrix} \\ &= \begin{bmatrix} f\big(x_k, u_k^{RL} + C_c x_k^c + D_c(y_k^r - y_k), w_k; \xi\big) \\ A_c x_k^c + B_c(y_k^r - y_k) \end{bmatrix} \\ &= \begin{bmatrix} f\big(x_k, u_k^{RL} + C_c x_k^c + D_c y_k^r - D_c h(x_k; \xi), w_k; \xi\big) \\ A_c x_k^c + B_c y_k^r - B_c h(x_k; \xi) \end{bmatrix} \\ &:= \mathcal{F}^{env}\big(\mathcal{X}_k^{env}, u_k^{RL}, w_k, y_k^r; \xi\big). \end{aligned} \tag{17}$$

The parameter vector ξ is fixed within each episode, and w_k and y_k^r are exogenous signals that are independent of the past states and inputs. Hence, (17) shows that, for a fixed realization of ξ, the distribution of the next augmented state $\mathcal{X}_{k+1}^{env}$ depends only on the current augmented state $\mathcal{X}_k^{env}$ and the learning-based input u_k^{RL}. The closed loop therefore satisfies the Markov property for each parameter realization, and constitutes an MDP whose state is $\mathcal{X}_k^{env}$ and whose action is u_k^{RL} [42], [43].

This result has two practical consequences. First, the RL algorithms and DR techniques developed for MDPs can be applied to the hybrid scheme without modification, because the MBC is simply absorbed into the environment. Second, the derivation uses only the state-space structure of the MBC in Assumption 2 and the general form of the plant (1) and (2); the result therefore holds for any such plant and controller, not only for the powertrain and the H_2 controller considered in this study. The family of these closed-loop MDPs indexed by ξ forms the basis of the policy optimization in Section IV-C.

C. Policy Optimization over a Family of MDPs

Under domain randomization, the training environment is no longer a single MDP. As shown in Section IV-B, each realization of ξ yields a closed-loop MDP, denoted by $\mathcal{M}(\xi)$, whose state is $\mathcal{X}_k^{env}$ and whose action is u_k^{RL}. The simulator can therefore be regarded as a family of MDPs $\mathcal{M} = \{\mathcal{M}(\xi)\}$ parameterized by the uncertain physical parameters, together with the distribution ρ_ξ from which the member used in each episode is randomly sampled [26], [27].

Let μ_θ be a policy parameterized by θ, and let $\tau_{1:T}$ denote a trajectory generated in $\mathcal{M}(\xi)$. The expected return under domain randomization is defined as

$$\begin{aligned} &J(\mu_\theta) \\ &\quad = \mathbb{E}_{\xi\sim\rho_\xi}\left[\mathbb{E}_{\tau\sim p(\tau|\mu_\theta,\xi)}\left[\sum_{i=1}^{T} \gamma^{i-1} r(s_i, a_i) \mid s_1, a_1; \mu_\theta\right]\right] \end{aligned} \tag{18}$$

where the likelihood of a trajectory $\tau = (s_1, a_1, s_2, \ldots, a_{T-1}, s_T)$ is denoted by $p(\tau|\mu_\theta, \xi)$ under the policy μ_θ. The learning objective is given by

$$\mu_\theta^* = \arg\max_{\mu_\theta} J(\mu_\theta). \tag{19}$$

The objective (18) evaluates a policy by its average return over the family rather than by its return in a single member. The distribution ρ_ξ thus specifies the range of variations against which robustness is sought: a wider ρ_ξ covers larger model discrepancies, at the cost of a harder learning problem and potentially more conservative policies, as discussed in Section I. The model-based assistance described in Section IV-A is introduced to alleviate this trade-off. The overall procedure is summarized in Algorithm 1.

V. Stability Analysis

This section analyzes the stability of the hybrid closed loop in continuous time, in which the MBC is designed. The analysis starts from the generalized plant description, so that the result applies to any internally stabilizing model-based controller rather than to a particular synthesis route.

Let the nominal plant used for the MBC design be written in generalized plant form:

$$\begin{aligned} \dot{x} &= A_p x + B_{p1} w + B_{p2} u \\ z &= C_{p1} x + D_{p12} u \\ y &= C_{p2} x + D_{p21} w \end{aligned} \tag{20}$$

where w collects the reference signal y^r and exogenous disturbances, z is the performance output, and y is the signal fed back to the controller; the tracking error $e = y^r - C_p x$

Algorithm 1 Model-Based Controller Assisted Domain-Randomized Training

Input: Environment E and trajectories pool $\mathcal{T}_{rand}$.
Output: Optimal actor policy.

1: Randomly initialize the actor's policy parameters and the critic parameters.
2: Design a MBC using the nominal system model.
3: **for** each episode **do**
4: $\xi \sim \rho_\xi$: Randomly sample a set of physical parameters.
5: $E \leftarrow \Xi(\xi)$: Reset the simulation environment
6: Generate rollout $\tau = (s_1, a_1, \ldots, s_T) \sim \mu_\theta + u_{MBC}$ with dynamics ξ and E.
7: $\mathcal{T}_{rand} \leftarrow \mathcal{T}_{rand} \cup \tau_i$
8: $u_{RL} \leftarrow \mu_\theta$: DRL control action from the policy.
9: $u = u_{RL} + u_{MBC}$: Integration of the data-driven policy u_{RL} and the physical model-based control u_{MBC}.
10: **for** each gradient step **do**
11: Update the critic parameters by minimizing the mean squared error.
12: Update the actor parameters using the sampled policy gradient.
13: $\theta \leftarrow \theta + \alpha_\theta \nabla_\theta J(\mu_\theta)$ with $\mathcal{T}_{rand}$ update.
14: $u = u_{RL} + u_{MBC}$: Update the hybrid control.
15: **end for**
16: **end for**

is recovered by an appropriate choice of C_{p2} and D_{p21}. The standard assumptions $D_{p11} = 0$ and $D_{p22} = 0$ are adopted. The MBC (7) and (8), written in continuous time, is

$$\dot{x}_c = A_K x_c + B_K y, \qquad u^{MBC} = C_K x_c + D_K y \tag{21}$$

The true plant departs from the nominal model because of the backlash nonlinearity and the parametric variations. This mismatch is treated as an additive unmodeled term $\Delta(t)$ acting on the state equation, rather than being embedded in the generalized plant as a linear fractional term:

$$\dot{x} = A_p x + B_{p1} w + B_{p2} u + \Delta \tag{22}$$

Substituting $y = C_{p2}x + D_{p21}w$ into (21) and applying the hybrid input (15) gives $u = C_K x_c + D_K(C_{p2}x + D_{p21}w) + u^{RL}$. With the augmented state $\mathcal{X} := [\,x^\top \;\; x_c^\top\,]^\top$, the closed loop becomes

$$\dot{\mathcal{X}} = A_{cl}\mathcal{X} + B_{cl} w + \tilde{B}_2 u^{RL} + \tilde{B}_\Delta \Delta \tag{23}$$

where

$$\begin{aligned} A_{cl} &= \begin{bmatrix} A_p + B_{p2} D_K C_{p2} & B_{p2} C_K \\ B_K C_{p2} & A_K \end{bmatrix}, \\ B_{cl} &= \begin{bmatrix} B_{p1} + B_{p2} D_K D_{p21} \\ B_K D_{p21} \end{bmatrix}, \\ \tilde{B}_2 &= \begin{bmatrix} B_{p2} \\ 0 \end{bmatrix}, \quad \tilde{B}_\Delta = \begin{bmatrix} I \\ 0 \end{bmatrix}. \end{aligned} \tag{24}$$

Assumption 3 (Bounded signals). The exogenous input, the learned action, and the model mismatch are bounded as $\|w(t)\| \le \bar{w}$, $\|u^{RL}(t)\| \le \bar{u}$, and $\|\Delta(t)\| \le \bar{\Delta}$.

The bound on u^{RL} holds by construction, since the actor terminates in a saturating output layer; the bound on Δ holds on the operating region because the dead-zone force is limited by the drivetrain stiffness and the backlash width, and the parametric variations lie in the ranges of domain randomization.

Proposition 1. Suppose that Assumption 3 holds. Since the MBC internally stabilizes the nominal plant, A_{cl} is stable, and for any $S = S^\top \succ 0$ the Lyapunov equation

$$A_{cl}^\top P + P A_{cl} = -S \tag{25}$$

has a unique solution $P = P^\top \succ 0$. Let

$$\begin{aligned} b &:= \|P B_{cl}\|\bar{w} + \|P\tilde{B}_2\|\bar{u} + \|P\tilde{B}_\Delta\|\bar{\Delta}, \\ \bar{\rho} &:= \frac{2b}{\lambda_{\min}(S)} \end{aligned} \tag{26}$$

and define the sublevel set of $V(\mathcal{X}) = \mathcal{X}^\top P \mathcal{X}$

$$\Omega := \big\{\mathcal{X} : V(\mathcal{X}) \le \lambda_{\max}(P)\,\bar{\rho}^{\,2}\big\}. \tag{27}$$

Then Ω is forward invariant, every trajectory asymptotically approaches Ω, and the closed loop state is ultimately bounded as

$$\limsup_{t\to\infty} \|\mathcal{X}(t)\| \le \sqrt{\kappa(P)}\;\bar{\rho}, \qquad \kappa(P) := \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}. \tag{28}$$

Proof: Let $V(\mathcal{X}) = \mathcal{X}^\top P \mathcal{X}$. Differentiating along (23) and using $P = P^\top$ to merge the two cross terms, then substituting (25), gives

$$\begin{aligned} \dot{V} &= \mathcal{X}^\top\big(A_{cl}^\top P + P A_{cl}\big)\mathcal{X} + 2\mathcal{X}^\top P\Big(B_{cl} w + \tilde{B}_2 u^{RL} + \tilde{B}_\Delta \Delta\Big) \\ &= -\mathcal{X}^\top S \mathcal{X} + 2\mathcal{X}^\top P\Big(B_{cl} w + \tilde{B}_2 u^{RL} + \tilde{B}_\Delta \Delta\Big) \end{aligned} \tag{29}$$

Applying the Rayleigh quotient to the first term, and the Cauchy–Schwarz inequality to each contribution in the second, yields

$$\begin{aligned} \dot{V} &\le -\lambda_{\min}(S)\|\mathcal{X}\|^2 + 2b\|\mathcal{X}\| \\ &= -\lambda_{\min}(S)\,\|\mathcal{X}\|\left(\|\mathcal{X}\| - \frac{2b}{\lambda_{\min}(S)}\right) \end{aligned} \tag{30}$$

Hence $\dot{V} < 0$ holds whenever $\|\mathcal{X}\| > \bar{\rho} := 2b/\lambda_{\min}(S)$.

The set Ω in (27) is the smallest sublevel set of V containing the ball $\mathcal{B} := \{\|\mathcal{X}\| \le \bar{\rho}\}$: for $\mathcal{X} \in \mathcal{B}$, the Rayleigh quotient gives $V(\mathcal{X}) \le \lambda_{\max}(P)\|\mathcal{X}\|^2 \le \lambda_{\max}(P)\bar{\rho}^{\,2}$, and the bound is attained on $\mathcal{B}$. Consequently $\mathcal{X} \notin \Omega$ implies $\mathcal{X} \notin \mathcal{B}$, that is, $\|\mathcal{X}\| > \bar{\rho}$, and therefore $\dot{V} < 0$ outside Ω. On the boundary of Ω one has $\lambda_{\max}(P)\|\mathcal{X}\|^2 \ge V(\mathcal{X}) = \lambda_{\max}(P)\bar{\rho}^{\,2}$, hence $\|\mathcal{X}\| \ge \bar{\rho}$ and $\dot{V} \le 0$; the trajectory cannot cross the boundary outward, so Ω is forward invariant [55].

Consequently V decreases until the trajectory enters the smallest sublevel set of V containing the ball $\{\|\mathcal{X}\| \le \bar{\rho}\}$, and cannot subsequently leave it, since $\dot{V} \le 0$ on its boundary. The bound (28) then follows from $\lambda_{\min}(P)\|\mathcal{X}\|^2 \le V(\mathcal{X}) \le \lambda_{\max}(P)\|\mathcal{X}\|^2$ [55]. ∎

Proposition 1 characterizes the region to which the hybrid closed loop is ultimately confined, and quantifies the contribution of the learning-based input to the Lyapunov decrease condition of the closed loop. The learning-based input enters $\bar{\rho}$ only through the single term $\|P\tilde{B}_2\|\bar{u}$, which is linear in the saturation bound $\bar{u}$. The learned policy therefore cannot destabilize the closed loop; its effect is limited to an enlargement of the ultimate bound (28) that is computable in advance from $\bar{u}$ and the design data. As $\bar{u} \to 0$ the expression reduces to the corresponding evaluation for the closed loop driven by w and Δ alone, so $\bar{u}$ acts as an explicit design parameter trading learning authority against the size of the guaranteed ultimate bound. The three contributions to $\bar{\rho}$ are separated by physical origin—the reference and disturbance channel, the learning-based input, and the model mismatch—and can be evaluated individually for a given design.

VI. Numerical Verifications

A. Nonlinear Powertrain Model

Numerical simulations are conducted on a nonlinear vehicle powertrain system with mechanical backlash and parametric uncertainty. The backlash introduces a dead-zone nonlinearity, resulting in mode-dependent dynamics that complicate vibration suppression. A schematic overview of the equivalent simple powertrain model is shown in Fig. 2. See the reference [36] [7] for more details on the powertrain model and the nominal parameters. The controlled output is defined as the vehicle body vibration $y_k = X_B$, and the control objective is to suppress vibration while tracking a reference signal under nonlinear and uncertain operating conditions.

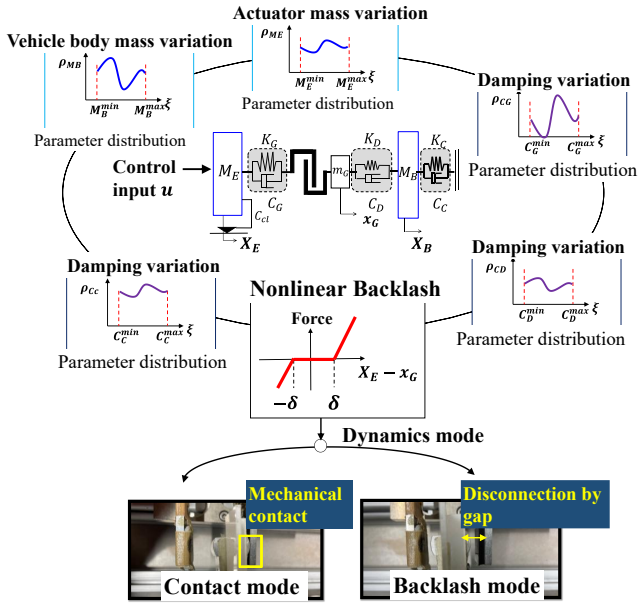


Fig. 2. Powertrain dynamics.

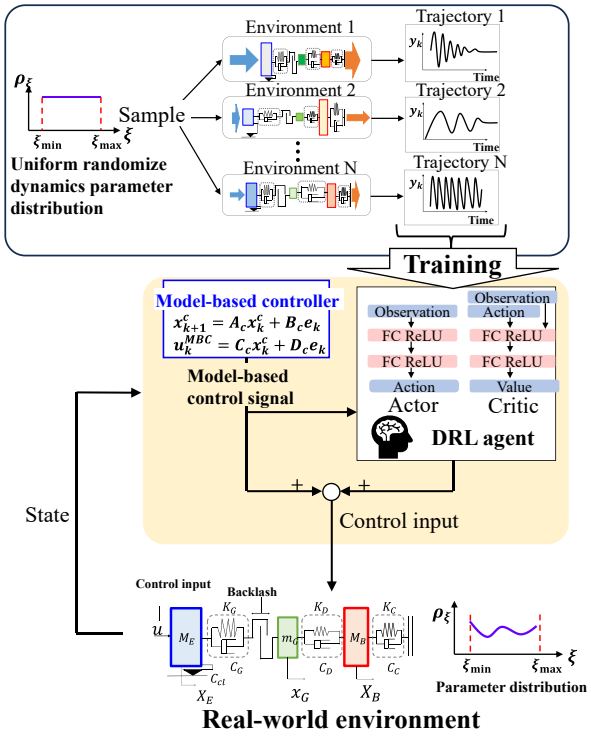


Fig. 3. Outline of domain randomized training.

B. Verification Settings

The verification settings consist of learning conditions and uncertainty conditions, which are summarized separately.

The hyperparameters and implementation settings used for training the learning-based controller are summarized in Table I. These settings are fixed throughout all simulations to ensure consistent evaluation. The proposed method is compared with three baselines. The first is the open-loop system without feedback control (No control). The second is the model-based controller alone (Only MBC), that is, the H_2 controller described in Section IV-A without the learning-based input. The third is domain-randomized DRL without model-based assistance (Only DR), in which the policy generates the entire control input. The proposed method and Only DR are trained under the same conditions listed in Table I, except for the components related to the model-based controller. Each learning-based method is trained with five random seeds. For application to powertrain control, a feedforward input is also involved [8], [56].

Parametric uncertainty is introduced through domain randomization. At the beginning of each episode, uncertain physical parameters are randomly sampled and kept fixed during the episode. The ranges of the randomized parameters, including the actuator mass $M_E \in \left[M_E^{min}, M_E^{max}\right]$, vehicle body mass $M_B \in \left[M_B^{min}, M_B^{max}\right]$, three damping coefficients $C_G \in \left[C_G^{min}, C_G^{max}\right]$, $C_D \in \left[C_D^{min}, C_D^{max}\right]$, $C_C \in \left[C_C^{min}, C_C^{max}\right]$, and backlash length $\delta \in \left[\delta^{min}, \delta^{max}\right]$, are summarized in Table II. Figure 3 outlines the training procedure.

The observation vector includes the reference signal, the controlled output, the tracking error and its integral and derivative, as well as the control input generated by the model-based controller. All observation signals are normalized. The reward function is defined as a quadratic cost penalizing tracking error and control effort, consistent with the problem formulation in Section II.

TABLE I
HYPERPARAMETERS FOR TD3.

Hyperparameter	***Value***
Sampling time	0.006 s
Critic learning rate	1.0×10^{-4}
Actor learning rate	1.0×10^{-4}
Discount factor γ	0.99
Number of the neurons in the hidden layers	64
Size of the mini-batch M	128
Size of the replay buffer	1.0×10^{5}
Horizon T	667
Network architecture	Fully connected + ReLU
Number of episodes	500
Target smoothing noise	$\mathrm{clip}(\mathcal{N}(0, 0.2), -0.5, 0.5)$
Exploration noise	Ornstein–Uhlenbeck action noise
Frequency of policy update with respect to critic update	2

C. Results and Discussion

Fig. 4 shows the episode reward during training over the five seeds. For Only DR (blue line), the learning curves vary considerably across seeds. In contrast, the proposed method (green line) yields learning curves with markedly smaller variation, suggesting that introducing the MBC stabilizes training. Because the early stage of training is dominated by trial-and-error exploration, a DRL agent that generates the entire control input is prone to destabilizing the closed loop, and the convergence of Only DR is therefore delayed compared with

TABLE II
RANDOMIZED PARAMETERS.

Parameter	*Range*
Damping coefficient C_D	Nominal value $\pm 50\%$: $[C_D^{min}, C_D^{max}] = [6.25, 18.75]$
Damping coefficient C_G	Nominal value $\pm 50\%$: $[C_G^{min}, C_G^{max}] = [18, 54]$
Damping coefficient C_C	Nominal value $\pm 50\%$: $[C_C^{min}, C_C^{max}] = [0.05, 0.15]$
Vehicle body mass M_B	Nominal value $\pm 50\%$: $[M_B^{min}, M_B^{max}] = [0.1160, 0.3480]$
Actuator mass M_E	Nominal value $\pm 50\%$: $[M_E^{min}, M_E^{max}] = [0.5200, 1.5600]$
Length of backlash δ	Nominal value $\pm 50\%$: $[\delta^{min}, \delta^{max}] = [0.0025, 0.0075]$
Steady value of the reference signal from 0 to 2 s y^r_{0-2s}	$\left[y^{r,min}_{0-2s}, y^{r,max}_{0-2s}\right] = [-0.01515, 0.030303]$ m
Steady value of the reference signal from 2 to 4 s y^r_{2-4s}	$\left[y^{r,min}_{2-4s}, y^{r,max}_{2-4s}\right] = [-0.01515, 0.030303]$ m

that of the proposed method. By running the MBC in parallel, the agent of the proposed method can avoid unproductive exploration, which improves convergence.

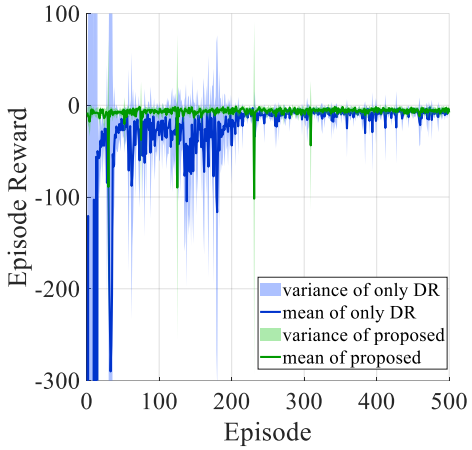


Fig. 4. Episode reward during training over five seeds.

Table III compares the control performance for the nominal model, which contains neither nonlinearities nor parametric uncertainty, using the 2-norm of the error between the reference and the controlled output. For the learning-based methods (the proposed method and Only DR), the mean and standard deviation over the five seeds are reported. All three closed-loop systems reduce the error substantially compared with No control, and Only MBC achieves the best performance. Since the model used for the controller design and the plant coincide, it is theoretically reasonable that the MBC performs best. In the proposed method, this nominal performance of the MBC serves as the foundation that assists the training of the DRL agent.

Next, robustness against plant uncertainty is examined. Figs. 5 and 6 show representative results when the power-train dynamics include parameter variations and the backlash nonlinearity. In these figures, multiple parameters, such as the

TABLE III
2-NORM OF THE TRACKING ERROR FOR THE NOMINAL MODEL (MEAN AND STANDARD DEVIATION OVER FIVE SEEDS)

Method	*Mean*	*Standard deviation*
Proposed method	0.7960	0.1458
Only DR	0.9788	0.1787
Only MBC	0.5890	–
No control	2.6940	–

vehicle mass, are set to the extremes of their ranges, with opposite combinations in the two cases. The upper and lower panels show the time histories of the vehicle body vibration and the control input, respectively. The ideal response (green line) contains abrupt changes in the driving force, under which vibration is most likely to be excited. Tables IV and V compare the 2-norms for the results in Figs. 5 and 6, respectively, where the mean and standard deviation are reported for the learning-based methods.

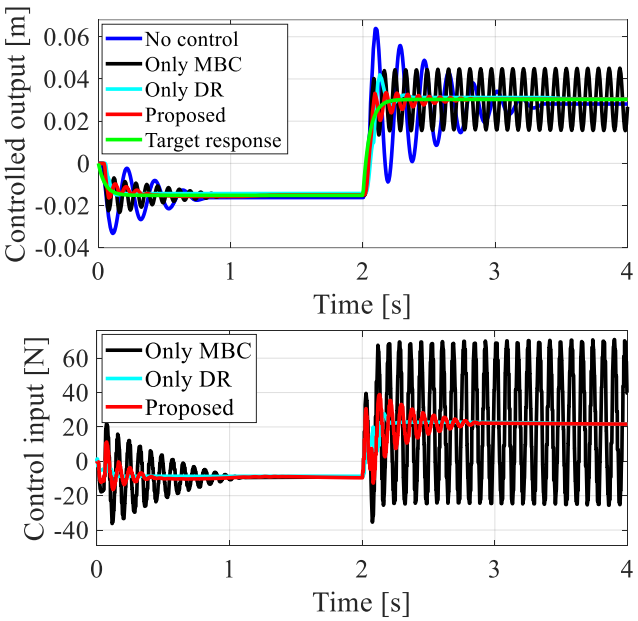


Fig. 5. Simulation result with $M_B = M_B^{min}$, $M_E = M_E^{min}$, $y^r_{0-2s} = y^{r,min}_{0-2s}$, $y^r_{2-4s} = y^{r,max}_{2-4s}$, $C_G = C_G^{max}$, $C_D = C_D^{min}$, $C_C = C_C^{min}$, and $\delta = \delta^{max}$.

TABLE IV
2-NORM OF THE TRACKING ERROR FOR FIG. 5 (MEAN AND STANDARD DEVIATION OVER FIVE SEEDS)

Method	*Mean*	*Standard deviation*
Proposed method	1.1863	0.1477
Only DR	1.3640	0.4619
Only MBC	3.3755	–
No control	4.1192	–

A notable finding of these robustness tests is that the proposed method (MBC + DRL with DR) reduces the variability with respect to both the training conditions and the plant

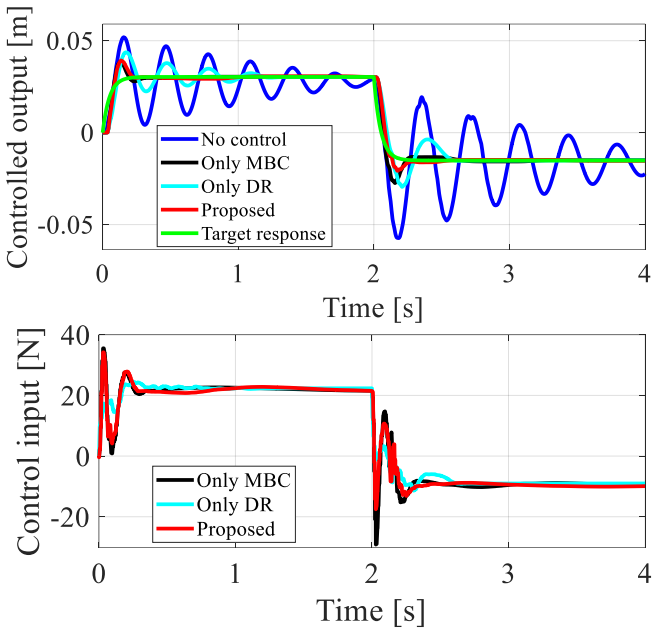


Fig. 6. Simulation result with $M_B = M_B^{max}$, $M_E = M_E^{max}$, $y^r_{0-2s} = y^{r,max}_{0-2s}$, $y^r_{2-4s} = y^{r,min}_{2-4s}$, $C_G = C_G^{min}$, $C_D = C_D^{max}$, $C_C = C_C^{max}$, and $\delta = \delta^{min}$.

TABLE V
2-NORM OF THE TRACKING ERROR FOR FIG. 6 (MEAN AND STANDARD DEVIATION OVER FIVE SEEDS)

Method	*Mean*	*Standard deviation*
Proposed method	1.6472	0.1802
Only DR	2.6016	1.0243
Only MBC	1.4891	–
No control	6.3969	–

variations. In Tables IV and V, the spread of the norm across seeds is substantially smaller for the proposed method than for Only DR. This indicates that the MBC can ensure the consistency of the policy against the stochastic uncertainty inherent in DRL with DR. DR is a major source of the diversity of the generated trajectories; the MBC is therefore particularly beneficial in training in which the exploration space tends to expand. The weak vibration suppression of Only DR observed in Fig. 6 is attributed to a suboptimal policy produced by excessive DR. The residual vibration that remains even with the proposed method in Fig. 5 suggests that the degree of DR in this study is too large. Nevertheless, the comparison of the mean 2-norms in Tables IV and V shows that the proposed method alleviates this conservativeness to some extent by introducing the MBC, because the MBC secures a baseline performance from the beginning of training. In other words, the proposed method is considered to enable learning focused on the uncertainty by efficiently restricting the space to be explored.

The comparison between the proposed method and Only MBC also indicates the robustness of the proposed method. Only MBC sometimes outperforms the proposed method, as in Table V, but its performance can also deteriorate severely, as in Table IV. Because the MBC is vulnerable to modeling errors, it cannot provide consistent robustness over diverse plants; the persistent vibration in Fig. 5 is a typical example. Although the proposed method shows somewhat insufficient performance in Fig. 5 and Table V, it can be regarded as broadly robust against various variations, owing to the improved generalization provided by DR.

To evaluate the robustness statistically, two further evaluations are conducted. The first is a Monte Carlo evaluation over plants obtained from 100 random samples. Since 100 samples are generated for each of the five training seeds, 500 closed-loop simulations are performed for each learning-based method. The mean and standard deviation of the 2-norm of the tracking error over these runs are summarized in Table VI.

The second is an extreme-parameter evaluation, which examines the performance at the boundary of the randomized ranges. Each method is evaluated at all $2^8 = 256$ vertices of the ranges listed in Table II, that is, combinations of the minimum and maximum values of the six physical parameters and the two reference levels. For the learning-based methods, this evaluation is repeated for each of the five trained policies, resulting in 1280 simulations per method. The mean and standard deviation of the 2-norm over these runs are summarized in Table VII. Whereas the Monte Carlo evaluation characterizes the typical performance over the training distribution, the extreme-parameter evaluation examines the performance under extreme parameter combinations at the boundary of this distribution, where controllers are most likely to degrade.

TABLE VI
2-NORM OF THE TRACKING ERROR IN THE MONTE CARLO EVALUATION (100 TRIALS FOR EACH OF FIVE SEEDS)

Method	*Mean*	*Standard deviation*
Proposed method	0.6998	0.3130
Only DR	0.9497	0.3393
Only MBC	0.6456	0.5002
No control	2.0325	1.2091

TABLE VII
2-NORM OF THE TRACKING ERROR IN THE EXTREME-PARAMETER EVALUATION (FIVE SEEDS)

Method	*Mean*	*Standard deviation*
Proposed method	1.0911	0.4104
Only DR	1.4778	0.6627
Only MBC	1.6698	1.3105
No control	3.5999	1.7109

The most important finding from these results is that the proposed method exhibits the most consistent robustness, with the smallest variability, across diverse plant variations. This is considered to result from the synergy between the baseline performance maintained by the MBC, which eases exploration during training, and the generalization ability provided by DRL. Only MBC achieves the smallest mean norm in Table VI, but its standard deviation is large, so that its performance varies with the plant realization. As Table VII

shows, this is because its performance deteriorates severely, particularly near the extreme parameter combinations. Only DR exhibits smaller variability than Only MBC but insufficient performance, because training does not progress sufficiently as DR enlarges the region to be explored. The proposed method achieves the best performance in Table VII; that is, DRL imparts robustness even to the plants for which the MBC deteriorates severely.

VII. Experimental Verifications

A. Experimental Setup and Conditions

The test bench used in the experiments is shown in Fig. 7(a). It reproduces the essential dynamics of a vehicle powertrain with backlash and has been employed in our previous studies [8], [56], [57]. The bench is a three-mass system with the structure shown in Fig. 2, driven by a linear motor whose thrust serves as the control input. The backlash is realized as an adjustable clearance between the leaf springs on both sides of the intermediate mass. Further details of the bench are given in [8], [56].

Fig. 7(b) illustrates the configuration of the closed-loop experiments. The four control systems compared in Section VI, namely No control, Only MBC, Only DR, and the proposed method, are implemented on a DSP (MIS Corporation) with a control period of $6\,\mathrm{ms}$, which coincides with the sampling time in Table I. The policies, i.e., multilayer perceptrons with two hidden layers of 64 neurons each, are executed on the DSP together with the MBC. The policies trained in simulation are deployed as they are, without any additional training or tuning on the hardware.

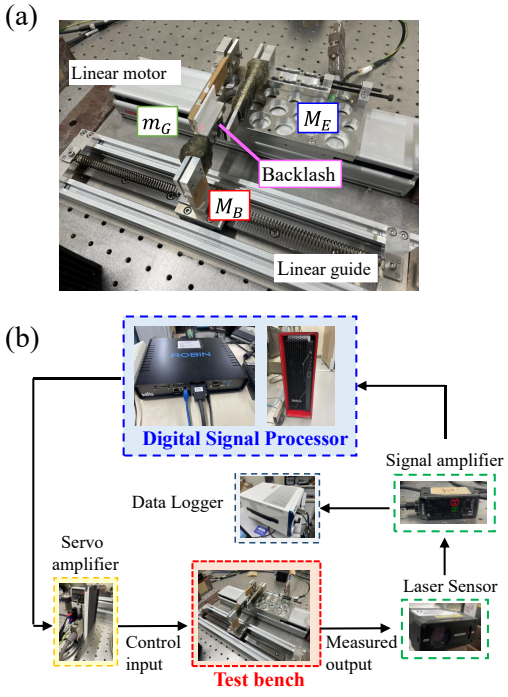


Fig. 7. Experimental setup. (a) Powertrain test bench. (b) Configuration of the closed-loop experiments.

The parameters used for training and for the design of the MBC are fixed at the nominal values used in Section VI, and the discrepancies are introduced on the hardware side. Two cases are considered. In Case 1, the backlash length is set to the value of $5\,\mathrm{mm}$, and a reference profile representing abrupt changes in the driving force is applied. In Case 2, the backlash length is increased by 50% to $7.5\,\mathrm{mm}$, which corresponds to the upper bound δ^{max} of the randomized range in Table II, and a reference profile different from that of Case 1 is applied. In addition, the damping characteristics of the bench, which are difficult to quantify, are expected to deviate from the nominal values owing to aging and other factors; both cases are therefore subject to such deviations.

For the learning-based methods, all five policies trained with different seeds are tested on the hardware, and the 2-norm of the tracking error is reported as the mean and standard deviation over the five policies.

B. Experimental Results and Discussion

The experiments examine whether the DRL policies trained in the simulation environment can be transferred to the physical system.

Figs. 8 and 9 show the experimental results in Case 1 and an enlarged view of them, respectively, and Fig. 10 shows the results in Case 2. The 2-norms of the tracking error in Cases 1 and 2 are summarized in Tables VIII and IX, respectively.

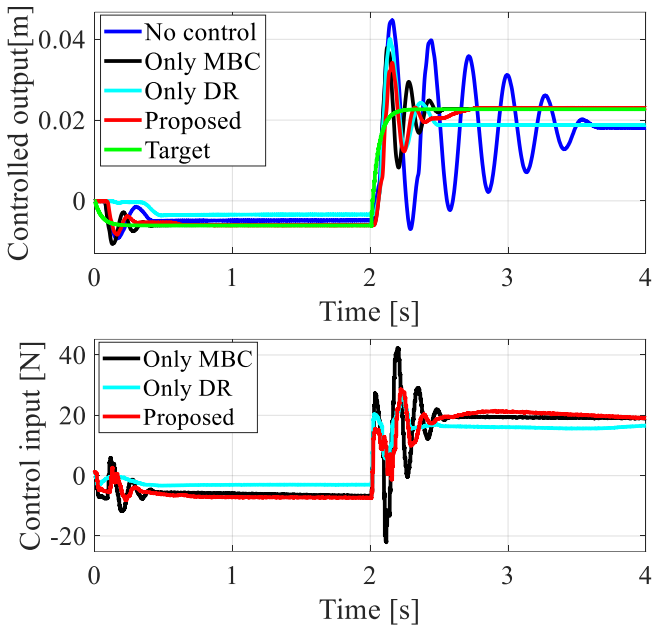


Fig. 8. Experimental results in Case 1.

As these results show, the proposed method (red line) achieves the most successful sim-to-real transfer, which is quantitatively supported by the smallest 2-norms in Tables VIII and IX. In Fig. 9, the proposed method suppresses the overshoot and the transient vibration immediately after the abrupt change in the driving force more effectively than the other closed-loop systems, and its response is the closest to the reference. The test bench is subject to the strong nonlinearity of backlash and, as noted in Section VII-A, to deviations of

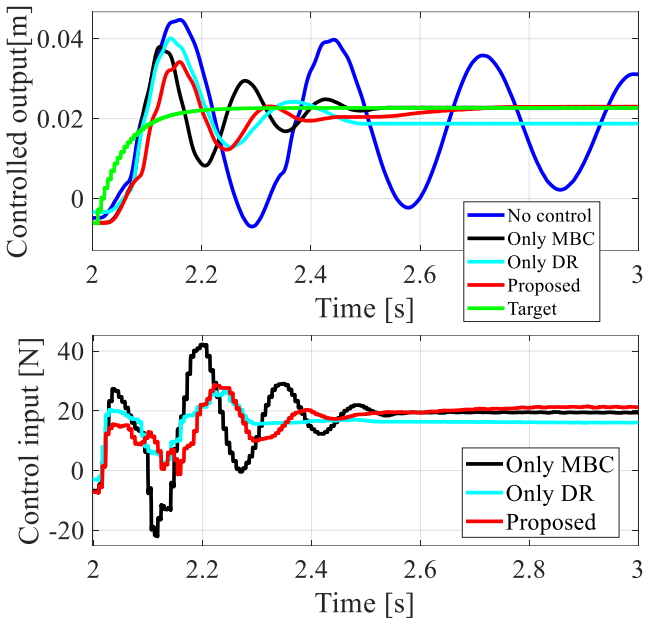


Fig. 9. Enlarged view of the results in Case 1.

the damping characteristics due to aging; the results show that these uncertainties are compensated for to some extent. In other words, the generalization ability that DRL adds to the MBC is confirmed on the physical system. In contrast, Only MBC leaves larger transient vibration, which stems from uncertainties not accounted for in the controller design. In particular, the shock caused by the collision in the backlash affects its response more prominently.

TABLE VIII
2-NORM OF THE TRACKING ERROR IN CASE 1 (MEAN AND STANDARD DEVIATION OVER FIVE SEEDS)

Method	*Mean*	*Standard deviation*
Proposed method	0.2339	0.0019
Only DR	0.4000	0.0021
Only MBC	0.2631	–
No control	0.7391	–

The response of Only DR (cyan line) is characterized by a conservative control input. Its large overshoot indicates that the compensation for backlash learned in simulation does not transfer well to the hardware. Although other uncertainties specific to the hardware, discussed later, also contribute to this result, factors arising from training cannot be ignored. When training relies on DRL alone, the exploration range enlarged by DR is considered too broad for the compensation of each individual uncertainty to be learned sufficiently. The proposed method, by contrast, secures the baseline performance from the outset and thereby promotes the acquisition of robustness focused on the uncertainties.

The control input highlights another benefit of introducing DRL, namely control efficiency. Compared with the proposed method and Only DR, Only MBC generates unnecessarily large control inputs. This inefficiency results from modeling errors, in other words, from a design that assumes a single fixed plant. In training with DR, by contrast, the agent learns how to adjust the control input while experiencing various plants, which is considered to lead to more efficient control. In particular, the DRL component of the proposed method corrects the excessive input of the MBC toward a more efficient one. The control input is analyzed in detail below.

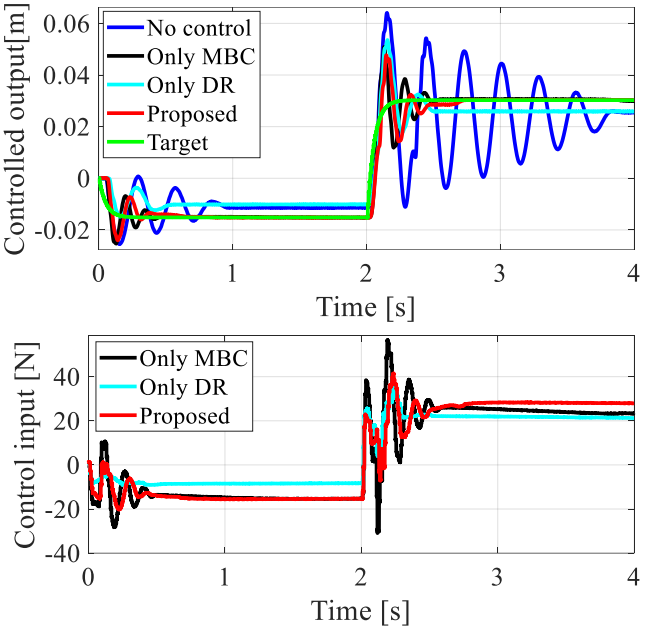


Fig. 10. Experimental results in Case 2.

TABLE IX
2-NORM OF THE TRACKING ERROR IN CASE 2 (MEAN AND STANDARD DEVIATION OVER FIVE SEEDS)

Method	*Mean*	*Standard deviation*
Proposed method	0.3605	0.0039
Only DR	0.5455	0.0018
Only MBC	0.3685	–
No control	1.1189	–

To examine the composition of the control input, Fig. 11 shows the contributions of the MBC and DRL separately. The MBC input (green line) is in phase with the total control input (red line), indicating that the MBC dominates the overall behavior. The DRL input (magenta line), in turn, corrects the excessive commands of the MBC and shapes them appropriately, that is, so as to compensate for the uncertainties. Immediately after 2.0 s, a clearance momentarily opens in the backlash. If the excessive input of the MBC were applied as it is, a shock would occur at the subsequent collision. To prevent this, the collision should be cushioned, that is, a soft landing is desired. Around 2.0 s, DRL computes an input in antiphase with that of the MBC, which decelerates the actuator and thereby compensates for the backlash. This complementary interaction between DRL and the MBC was identified through a numerical analysis in our previous work

[36]; one of the novelties of this study is that its existence is confirmed experimentally.

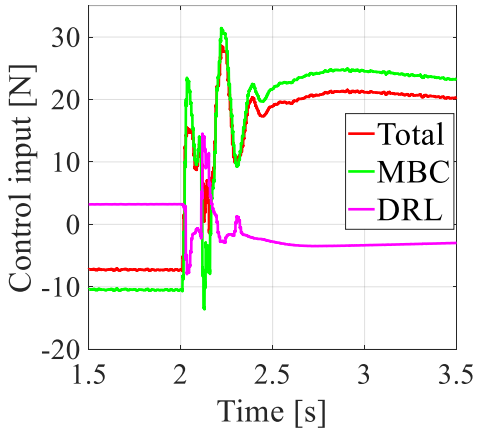


Fig. 11. Decomposition of the control input of the proposed method into the contributions of the MBC and DRL.

The experimental results also reveal room for improvement in control performance. In particular, the proposed method does not suppress the overshoot as effectively as in the simulation; compared with the ideal response (green line), the compensation for backlash is still insufficient. Moreover, unlike the trend in the simulation, the standard deviation of the proposed method in Table IX is larger than that of Only DR. These observations are attributed to uncertainties in the hardware that are not considered in the training environment; specifically, additional friction in the actuator and nonlinear characteristics of the springs may be present. The steady-state error of Only DR suggests a dominant influence of friction or similar effects. With respect to the steady-state error, introducing an MBC that explicitly includes integral action is effective, as demonstrated by the proposed method. On the other hand, the overshoot and residual vibration caused by backlash may be attributed to the degradation of the MBC performance. Accordingly, one limitation of this study is its vulnerability to uncertainties not assumed in the training environment. Possible countermeasures include improving the robustness of the MBC and refining the training environment to reduce the sim-to-real gap.

VIII. Conclusion

This paper presented MBCADR, a framework that integrates model-based control into domain-randomized DRL for robust vibration suppression of powertrains with backlash. A residual policy trained by TD3 under randomized dynamics compensates for the model mismatch of an H_2 controller designed from the nominal model. Domain-randomized training was formulated as policy optimization over a family of MDPs, and a Lyapunov-based analysis provided a forward-invariant set and an ultimate bound for the hybrid closed loop. Monte Carlo and extreme-parameter evaluations over five training seeds showed that the proposed method yields the smallest performance variability across plant realizations. Experiments on a powertrain test bench confirmed that the policies trained in simulation transfer to the hardware without further tuning. Future work includes applying the framework to other control systems and further mitigating the conservativeness induced by domain randomization.

Acknowledgment

During the preparation of this work, the author(s) used Claude Opus 5.5 (Anthropic) for language editing, drafting, and assistance with manuscript preparation. After using this tool/service, the author(s) reviewed, verified, and edited the content as needed and take(s) full responsibility for the content of the published article.

References

[1] F. Wang, T. Wu, Y. Ni, P. Ye, Y. Cai, J. Guo, and C. Wang, "Torsional oscillation suppression-oriented torque compensate control for regenerative braking of electric powertrain based on mixed logic dynamic model," *Mechanical Systems and Signal Processing*, vol. 190, p. 110114, 2023.

[2] F. Wang, P. Ye, X. Xu, Y. Cai, S. Ni, and H. Que, "Novel regenerative braking method for transient torsional oscillation suppression of planetary-gear electrical powertrain," *Mechanical Systems and Signal Processing*, vol. 163, p. 108187, 2022.

[3] C. Wu, K. Guo, J. Sun, Y. Liu, and D. Zheng, "Active vibration control in robotic grinding using six-axis acceleration feedback," *Mechanical Systems and Signal Processing*, vol. 214, p. 111379, 2024.

[4] S. Zaare and M. R. Soltanpour, "Adaptive fuzzy global coupled non-singular fast terminal sliding mode control of n-rigid-link elastic-joint robot manipulators in presence of uncertainties," *Mechanical Systems and Signal Processing*, vol. 163, p. 108165, 2022.

[5] Y. Yang, C. Liu, T. Zhang, X. Zhou, and J. Li, "Event-triggered composite sliding mode anti-sway control for tower crane systems and experimental verification," *Mechanical Systems and Signal Processing*, vol. 230, p. 112578, 2025.

[6] Z. Xu, G. Qi, Q. Liu, and J. Yao, "ESO-based adaptive full state constraint control of uncertain systems and its application to hydraulic servo systems," *Mechanical Systems and Signal Processing*, vol. 167, p. 108560, 2022.

[7] H. Yonezawa, A. Yonezawa, T. Hatano, S. Hiramatsu, C. Nishidome, and I. Kajiwara, "Fuzzy-reasoning-based robust vibration controller for drivetrain mechanism with various control input updating timings," *Mechanism and Machine Theory*, vol. 175, p. 104957, 2022.

[8] H. Yonezawa, A. Yonezawa, and I. Kajiwara, "Experimental validation of adaptive grey wolf optimizer-based powertrain vibration control with backlash handling," *Mechanism and Machine Theory*, vol. 203, p. 105825, 2024.

[9] R. E. Kalman, "On the general theory of control systems," *IFAC Proceedings Volumes*, vol. 1, pp. 491–502, 1960.

[10] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1996.

[11] S. Xu, X. Liu, Y. Wang, Z. Sun, J. Wu, and Y. Shi, "Frequency shaping-based H_∞ control for active pneumatic vibration isolation with input voltage saturation," *Mechanical Systems and Signal Processing*, vol. 220, p. 111705, 2024.

[12] L. Chen, X. Liu, M. Wu, K. Hirota, and W. Pedrycz, "Robust h-infinity control of feeding speed in coal seam drilling process with uncertain hardness," *IEEE Transactions on Control Systems Technology*, vol. 34, no. 1, pp. 381–394, 2026.

[13] Y. Li, J. Zhao, Y. Jiang, and W. Liu, "H2/hinfinity multiobjective robust control with closed-loop pole constraints for ipt system," *IEEE Transactions on Control Systems Technology*, vol. 34, no. 6, pp. 3212–3227, 2026.

[14] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3803–3810.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[18] J. K. Viswanadhapalli, V. K. Elumalai, S. S., S. Shah, and D. Mahajan, "Deep reinforcement learning with reward shaping for tracking control and vibration suppression of flexible link manipulator," *Applied Soft Computing*, vol. 152, p. 110756, 2024.

[19] M. H. Ahmed, A. AboHussien, A. El-Shafei, A. M. Darwish, and A. H. Abdel-Gawad, "Active control of flexible rotors using deep reinforcement learning with application of multi-actor-critic deep deterministic policy gradient," *Engineering Applications of Artificial Intelligence*, vol. 124, p. 106593, 2023.

[20] C. Wang, W. Cheng, H. Zhang, W. Dou, and J. Chen, "An immune optimization deep reinforcement learning control method used for magnetorheological elastomer vibration absorber," *Engineering Applications of Artificial Intelligence*, vol. 137, p. 109108, 2024.

[21] Z. cheng Qiu, Y. hong Liu, and X. min Zhang, "Reinforcement learning vibration control and trajectory planning optimization of translational flexible hinged plate system," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108630, 2024.

[22] M. Malmir, J. Josifovski, N. Klarmann, and A. Knoll, "Diarel: Reinforcement learning with disturbance awareness for robust sim2real policy transfer in robot control," *IEEE Transactions on Control Systems Technology*, vol. 34, no. 2, pp. 1037–1043, 2026.

[23] A. Rehan, M. Chehadeh, I. Boiko, and Y. Zweiri, "Optimal nonparametric autotuning of pid controllers in unmanned aerial vehicles based on reinforcement learning and homogeneity," *IEEE Transactions on Control Systems Technology*, vol. 34, no. 5, pp. 2648–2663, 2026.

[24] G. Chen, Z. Chen, L. Wang, and W. Zhang, "Deep deterministic policy gradient and active disturbance rejection controller based coordinated control for gearshift manipulator of driving robot," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105586, 2023.

[25] T. Slawik, B. Wehbe, L. Christensen, and F. Kirchner, "Deep reinforcement learning for path-following control of an autonomous surface vehicle using domain randomization," *IFAC-PapersOnLine*, vol. 58, no. 20, pp. 21–26, 2024.

[26] J. Zhang, C. Zhao, and J. Ding, "Deep reinforcement learning with domain randomization for overhead crane control with payload mass variations," *Control Engineering Practice*, vol. 141, p. 105689, 2023.

[27] X. Chen, J. Hu, C. Jin, L. Li, and L. Wang, "Understanding domain randomization for sim-to-real transfer," *arXiv preprint arXiv:2110.03239*, 2022.

[28] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," *arXiv preprint arXiv:1806.07851*, 2018.

[29] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2811–2817.

[30] G. A. Castillo, H. Lodha, and A. Hereid, "Learning terrain-aware bipedal locomotion via reduced-dimensional perceptual representations," *IEEE Transactions on Control Systems Technology*, vol. 34, no. 3, pp. 1149–1161, 2026.

[31] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems*, 2020.

[32] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath, "Berkeley humanoid: A research platform for learning-based control," *arXiv preprint arXiv:2407.21781*, 2024.

[33] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing Humanoid Locomotion: Mastering Challenging Terrains with Denoising World Model Learning," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[34] B. Yu and T. Lee, "Equivariant reinforcement learning frameworks for quadrotor low-level control," *IEEE Transactions on Control Systems Technology*, vol. 34, no. 1, pp. 86–99, 2026.

[35] Y. Fan, H. Dong, X. Zhao, and P. Denissenko, "Path-following control of unmanned underwater vehicle based on an improved td3 deep reinforcement learning," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 5, pp. 1904–1919, 2024.

[36] H. Yonezawa, A. Yonezawa, and I. Kajiwara, "Model-based controller assisted domain randomization for transient vibration suppression of nonlinear powertrain system with parametric uncertainty," *Mechanical Systems and Signal Processing*, vol. 241, p. 113570, 2025.

[37] O. Nachum, M. Ahn, H. Ponte, S. S. Gu, and V. Kumar, "Multi-agent manipulation via locomotion using hierarchical Sim2Real," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100, 2019, pp. 110–121.

[38] F. Ramos, R. Possas, and D. Fox, "BayesSim: Adaptive domain randomization via probabilistic inference for robotics simulators," in *Robotics: Science and Systems XV*, 2019.

[39] Y. Cheng, P. Zhao, F. Wang, D. J. Block, and N. Hovakimyan, "Improving the robustness of reinforcement learning policies with $\updownarrow_1$ adaptive control," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6574–6581, 2022.

[40] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6023–6029.

[41] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, "Residual policy learning," *arXiv preprint arXiv:1812.06298*, 2018.

[42] I. Koryakovskiy, M. Kudruss, H. Vallery, R. Babuška, and W. Caarls, "Model-plant mismatch compensation using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2471–2477, 2018.

[43] Y. Okawa, T. Sasaki, and H. Iwane, "Control approach combining reinforcement learning and model-based control," in *2019 12th Asian Control Conference (ASCC)*, 2019, pp. 1419–1424.

[44] J. Park, B. Kim, and S. Han, "Reinforcement learning with model-based assistance for shape control in sendzimir rolling mills," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 4, pp. 1867–1874, 2023.

[45] H. Cao, Y. Mao, L. Sha, and M. Caccamo, "Physics-model-regulated deep reinforcement learning towards safety & stability guarantees," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 8306–8311.

[46] Y. T. Liu, E. Price, M. J. Black, and A. Ahmad, "Deep residual reinforcement learning based autonomous blimp control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 12 566–12 573.

[47] Y. Zuo, Y. T. Liu, and A. Ahmad, "Autonomous blimp control via h_∞ robust deep residual reinforcement learning," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–8.

[48] J. Kim, J. Kwon, H. S. Kim, T. Seo, and H.-T. Seo, "Stability-aware residual reinforcement learning framework for robotic manipulator disturbance compensation," *arXiv preprint arXiv:2609.21307*, 2026.

[49] Y. Yan, T. Egle, C. Ott, and D. Lee, "Efficiently learning robust torque-based locomotion through reinforcement with model-based supervision," *IEEE Robotics and Automation Letters*, vol. 11, no. 4, pp. 4155–4162, 2026.

[50] H. Yonezawa, A. Yonezawa, and I. Kajiwara, "Model-based assisted domain-randomized reinforcement learning for robust powertrain vibration control," in *2026 IEEE Conference on Control Technology and Applications (CCTA)*, Vancouver, Canada, 2026, pp. 186–192.

[51] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*, ser. ICML'14. JMLR.org, 2014, pp. I–387–I–395.

[52] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2019.

[53] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *35th International Conference on Machine Learning (ICML)*, 2018, pp. 1587–1596.

[54] M. Chilali and P. Gahinet, "H_∞ design with pole placement constraints: an LMI approach," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 358–367, 1996.

[55] H. K. Khalil, *Nonlinear systems*, 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 2002.

[56] H. Yonezawa, I. Kajiwara, S. Sato, C. Nishidome, M. Sakata, T. Hatano, and S. Hiramatsu, "Vibration control of automotive drive system with nonlinear gear backlash," *ASME. Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 12, p. 121002, 2019.

[57] H. Yonezawa, I. Kajiwara, C. Nishidome, S. Hiramatsu, M. Sakata, and T. Hatano, "Vibration control of automotive drive system with backlash considering control period constraint," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 13, no. 1, p. JAMDSM0018, 2019.