

Physics-Informed Residual Neural Dynamics for Data-Efficient Predictive Control in Legged Robots

Sanjoy Kumar*

Mathematics and Statistics
Texas Tech University
sanjkuma@ttu.edu

Shovan Kumar Paul

Computer Science and Engineering
R.P. Shaha University
shovan.cse@rpsu.edu.bd

Lula Babole Alphonse-Roger

Mathematics, Statistics and Computer Science
University of Kinshasa
lulababole@gmail.com

Abstract—Model predictive control of floating-base legged systems requires dynamic formulations that balance real-time computational tractability with the capacity to capture unmodeled electro-mechanical and contact phenomena. While black-box neural dynamics can approximate nonlinear discrepancies, they demand extensive uniform state coverage, accumulate compounding drift over multi-step integration horizons, and frequently violate fundamental contact invariants. This paper presents Physics-Informed Neural Dynamics (PIND), a continuous-time residual framework tailored for data-efficient learning and predictive control of legged locomotion. PIND augments an analytical multi-body rigid-body model with two structured neural modules: an internal drivetrain residual strictly projected onto the actuated joint manifold, and a continuous contact compliance model. The network is trained using a composite physics-informed loss that regularizes generalized momentum balances, friction cone geometric admissibility, and unilateral contact complementarity. We prove an analytical trajectory error bound demonstrating that state forecasting error is explicitly bounded by the dynamic momentum residual scaled by the minimum eigenvalue of the generalized mass matrix. The learned continuous vector field is embedded into an interior-point Sequential Quadratic Programming (SQP) Nonlinear Model Predictive Control (NMPC) loop. Evaluated on the Unitree Go1 quadruped in MuJoCo subject to measurement noise, low training budgets (1%–100%), and parametric shifts in payload (+48%) and terrain friction ($\mu = 0.2$), PIND achieves stable multi-step forecasting with 10% of the data required by black-box neural baselines and preserves dynamic feasibility for real-time 50 Hz execution.

Index Terms—Legged locomotion, physics-informed machine learning, neural ordinary differential equations, nonlinear model predictive control, contact mechanics.

I. INTRODUCTION

Model-based predictive control architectures, such as Whole-Body Control (WBC) and Model Predictive Control (MPC) based on Single Rigid Body Models (SRBM) or full Rigid Body Dynamics (RBD), form the algorithmic foundation of modern legged locomotion [1], [2]. Reduced template models enable convex real-time optimization but neglect leg swing

inertia, motor back-EMF, transmission friction hysteresis, and compliant ground deformation [3]. Conversely, full rigid-body models incorporate kinematic couplings but remain susceptible to parametric errors and unmodeled joint drivetrain compliance.

Data-driven techniques have been explored to bridge these discrepancies. Deep reinforcement learning policies trained in randomized simulations achieve agile behaviors [4], [5], [13], yet operate as black-box controllers lacking formal constraint satisfaction or sample efficiency when fine-tuning directly on hardware. System identification via Neural Ordinary Differential Equations (Neural ODEs) [7] offers a continuous-time formulation for dynamic systems. However, purely data-driven ODEs lack physical inductive biases; when integrated recursively across long prediction horizons, unconstrained models exhibit compounding prediction drift and generate unphysical behaviors, such as terrain penetration or contact wrenches exceeding Coulomb friction limits [9], [10].

Physics-Informed Machine Learning (PIML) embeds governing physical principles into network architectures or loss functions [11], [15]. While Hamiltonian Neural Networks (HNNs) [9], Lagrangian Neural Networks (LNNs) [10], and Deep Lagrangian Networks (DeLaN) [16] enforce energy conservation or Euler-Lagrange equations, their standard formulations assume smooth, continuous, and conservative systems. Extending these paradigms to legged locomotion presents major mathematical challenges: legged robots are underactuated, subject to motor torque saturation, and governed by non-smooth, hybrid contact mechanics [12]. Recent residual dynamics approaches for quadrupeds typically learn unconstrained corrections in state space [19], [20], which fails to preserve contact geometry and yields ill-conditioned sensitivities during numerical trajectory optimization.

In this work, we propose **Physics-Informed Neural Dynamics (PIND)**, a structured continuous-time learning and control framework for legged locomotion. Rather than learning unconstrained transitions, PIND couples nominal multi-body dynamics with two structured neural components: an internal drivetrain residual projected strictly onto the actuated joint manifold, and a continuous contact compliance network. The training objective regularizes Newton-Euler momentum balance, friction cone geometry, and unilateral contact complementarity.

The primary contributions of this paper are:

- 1) **Structured Hybrid Continuous Architecture:** We es-

Sanjoy Kumar, PhD Student & Research Assistant, Department of Mathematics and Statistics, Texas Tech University. (email: sanjkuma@ttu.edu)

Shovan Kumar Paul, Assistant Professor, Department of Computer Science and Engineering, R.P. Shaha University. (email: shovan.cse@rpsu.edu.bd)

Lula Babole Alphonse-Roger, Professor, Department of Mathematics, Statistics and Computer Science, Faculty of Science and Technology, University of Kinshasa, Democratic Republic of Congo. (email: lulababole@gmail.com)

Source code, evaluation scripts are available at: <https://github.com/imsanjoykb/PIND-Quadruped>

tablish an underactuation-projected residual formulation that prevents the unactuated floating base from producing fictitious internal wrenches while capturing joint-level drive dynamics.

- 2) **Physics-Regularized Optimization Objective:** We develop a composite loss penalizing multi-body momentum imbalances, friction cone violations, and contact complementarity, regularizing state forecasting in low-data regimes.
- 3) **Momentum-Coupled Prediction Error Bound:** We provide an analytical proof demonstrating that open-loop state prediction error is bounded by the momentum residual scaled by the minimum eigenvalue of the generalized mass matrix.
- 4) **Predictive Control Integration with Computational Profiling:** We embed the learned vector field into an SQP-based NMPC formulation, analyze gradient conditioning, and quantify execution times to validate real-time 50 Hz feasibility.

II. RELATED WORK

A. Learned Dynamics and Continuous Neural Differential Equations

Data-driven dynamics learning approximates transitions $x_{t+1} = f_\theta(x_t, u_t)$. While Gaussian Processes (GPs) provide uncertainty metrics, their cubic computational complexity $\mathcal{O}(N^3)$ precludes deployment in high-frequency, contact-rich locomotion [14]. Deep neural networks, LSTMs, and Neural ODEs [6], [7] provide higher representational capacity, but unconstrained models remain prone to error accumulation over long rollout horizons. In multi-contact settings, unregularized networks frequently extrapolate into physically impossible state regimes outside the training distribution [9].

B. Physics-Informed Mechanical Models

Incorporating physical priors into machine learning models constrains hypothesis spaces [11], [15]. Symplectic and Lagrangian architectures preserve conservative invariants for unconstrained robotic arms [10], [16]. However, legged locomotion is inherently non-conservative, dissipative, and subject to intermittent unilateral ground reactions. Prior residual quadruped learning frameworks [13], [19] either limit identification to scalar lumped parameters or train unconstrained residual MLPs that neglect the geometric boundaries of Coulomb friction cones.

C. Model Predictive Control for Legged Locomotion

State-of-the-art quadrupedal predictive control relies on reduced models (e.g., SRBM) to formulate convex quadratic programs (QPs) over ground reaction forces [1], [2]. While computationally efficient, these models disregard limb inertia and joint dynamics. Nonlinear MPC (NMPC) over full rigid-body dynamics [22] captures complete multi-body coupling but remains sensitive to unmodeled drivetrain friction and terrain elasticity. Learning-based MPC incorporates data-driven

residual models into trajectory optimization [20], [21]; maintaining physical admissibility and well-conditioned sensitivities within learned dynamic surrogates is essential for numerical convergence in real-time execution.

III. PROBLEM FORMULATION

A. Underactuated Multi-Body Dynamics

We consider a legged robot modeled as an underactuated floating-base multi-body system with $n = 6 + n_j$ degrees of freedom ($n_j = 12$ actuated joints). The generalized coordinate vector is:

$$q = [p_b^\top \quad \phi_b^\top \quad q_j^\top]^\top \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^{12}, \quad (1)$$

where $p_b \in \mathbb{R}^3$ is base position, $\phi_b \in SO(3)$ is base orientation, and $q_j \in \mathbb{R}^{12}$ represents actuated joint angles. Generalized velocity is $\nu = [v_b^\top, \omega_b^\top, \dot{q}_j^\top]^\top \in \mathbb{R}^n$.

The equations of motion subject to intermittent foot contacts are:

$$M(q)\dot{\nu} + C(q, \nu)\nu + g(q) = B\tau + \sum_{i=1}^4 J_{c,i}^\top(q)\lambda_i + \tau_{\text{ext}} + \tau_{\text{dist}}, \quad (2)$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the generalized mass matrix, $C(q, \nu) \in \mathbb{R}^{n \times n}$ captures Coriolis and centrifugal effects, $g(q) \in \mathbb{R}^n$ denotes gravitational forces, $B = [0_{12 \times 6}, I_{12 \times 12}]^\top \in \mathbb{R}^{n \times 12}$ is the actuation selection matrix, and $\tau \in \mathbb{R}^{12}$ is the commanded motor torque vector. The contact wrench at foot i is denoted by $\lambda_i = [f_{i,x}, f_{i,y}, f_{i,z}]^\top \in \mathbb{R}^3$, mapped via the contact Jacobian $J_{c,i}(q) = \frac{\partial p_{c,i}(q)}{\partial q} \in \mathbb{R}^{3 \times n}$, where $p_{c,i}(q)$ is the position of foot i . Drivetrain unmodeled dynamics and external disturbances are represented by τ_{ext} and τ_{dist} .

B. Contact Mechanics and Complementarity

For foot i in contact with terrain characterized by friction coefficient μ , the contact wrench must lie within the Coulomb friction cone:

$$\mathcal{F}_i = \left\{ \lambda_i \in \mathbb{R}^3 \mid \lambda_{i,z} \geq 0, \sqrt{\lambda_{i,x}^2 + \lambda_{i,y}^2} \leq \mu \lambda_{i,z} \right\}. \quad (3)$$

Kinematic non-penetration of foot clearance above the ground surface $h(p_{c,i}^{x,y})$ requires:

$$\phi_{c,i}(q) := p_{c,i}^z(q) - h(p_{c,i}^{x,y}(q)) \geq 0. \quad (4)$$

Under rigid-body contact mechanics, unilateral contact satisfies the complementarity condition:

$$0 \leq \phi_{c,i}(q) \perp \lambda_{i,z} \geq 0 \implies \phi_{c,i}(q)\lambda_{i,z} = 0. \quad (5)$$

C. Learning Objective

Let state be $x(t) = [q(t)^\top, \nu(t)^\top]^\top \in \mathcal{X} \subset \mathbb{R}^{2n}$ and control input be $u(t) = \tau(t) \in \mathcal{U} \subset \mathbb{R}^{12}$. Given a trajectory training dataset $\mathcal{D} = \{(x_k, u_k, x_{k+1})\}_{k=1}^N$, our objective is to learn a parameterized continuous vector field:

$$\dot{x}(t) = f_\theta(x(t), u(t)), \quad (6)$$

such that open-loop multi-step prediction error $\|x(t+H) - \hat{x}(t+H)\|$ is minimized across horizon H , physical invariants

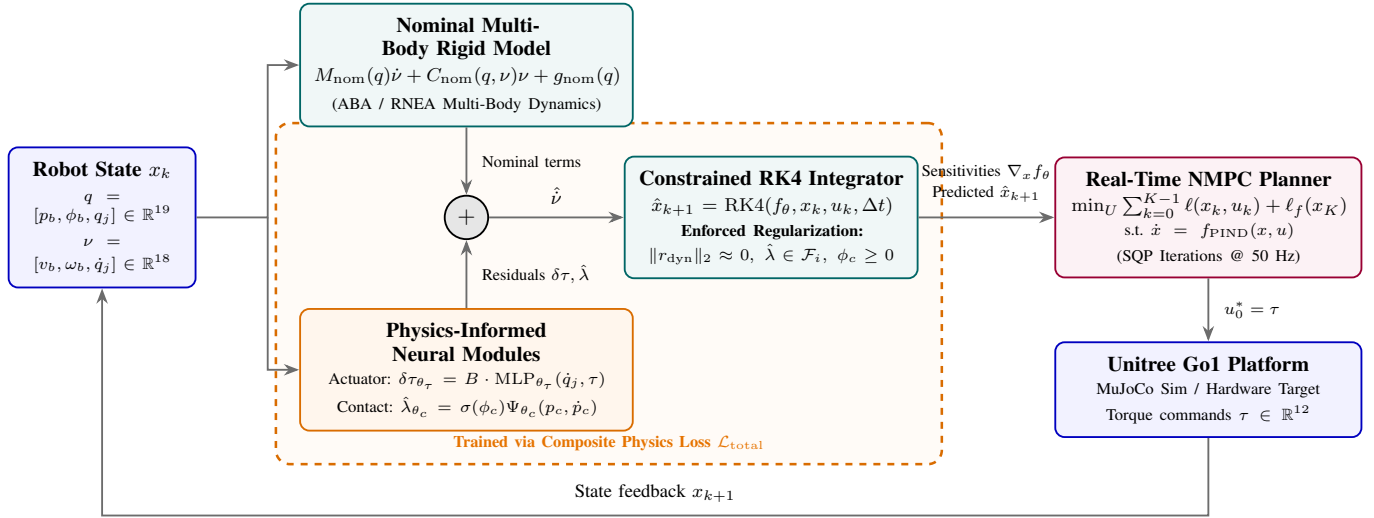


Fig. 1. System architecture of the Physics-Informed Neural Dynamics (PIND) model embedded within a real-time NMPC loop. Nominal multi-body dynamics are coupled with underactuation-projected drivetrain and contact residuals, trained using momentum balance and contact regularizers.

TABLE I
TAXONOMY OF RELATED RESEARCH IN LEARNED ROBOT DYNAMICS AND PREDICTIVE CONTROL

Paper	Year	Platform	Method	Physical Knowledge	Control	Primary Limitation
Raissi et al. [11]	2019	General PDEs	PINN	Differential operator residuals	None	Offline; smooth continuous systems
Greydanus et al. [9]	2019	Ideal Pendulum	HNN	Symplectic structure	None	Conservative systems only; unactuated
Lutter et al. [16]	2019	Manipulator	DeLaN	Euler-Lagrange structure	Inverse Dyn.	Continuous dynamics; contact-free
Hwangbo et al. [13]	2019	ANYmal	Actuator Net + RL	Kinematics + SEA model	RL Policy	Requires large-scale simulation rollouts
Zhong et al. [12]	2020	Cart-Pole, Pendulum	Neural ODE (Symplectic)	Port-Hamiltonian	Energy Shaping	Evaluated on low-dimensional benchmarks
Lee et al. [4]	2020	ANYmal	Teacher-Student RL	Kinematic priors	Blind Locom.	High sample complexity; black-box policy
Miki et al. [5]	2022	ANYmal	Elevation Map + RL	Kinematics + Depth	RL Policy	Model-free; lacks constraint guarantees
Di Carlo et al. [2]	2018	MIT Cheetah 3	Convex MPC	Single Rigid Body Dynamics	Convex QP	Unmodeled limb inertia and drive damping
Grandia et al. [22]	2019	ANYmal	Nonlinear MPC	Full Rigid Body Dynamics	SQP / WBC	Sensitive to unmodeled contact compliance
PIND (Proposed)	2026	Unitree Go1	Physics-Informed ODE	Multi-body ODE + Contact + Friction	Nonlinear MPC	Continuous contact relaxation required

are regularized out-of-distribution, and sensitivity evaluations $\nabla_x f_\theta$ are well-conditioned for predictive control.

IV. PHYSICS-INFORMED NEURAL DYNAMICS (PIND)

A. Architectural Formulation

We formulate the generalized acceleration using an analytical multi-body model augmented with structured neural components (Fig. 1):

$$\dot{\nu} = M_{\text{nom}}^{-1}(q) \left[B\tau - C_{\text{nom}}(q, \nu)\nu - g_{\text{nom}}(q) + \sum_{i=1}^4 J_{c,i}^\top(q) \hat{\lambda}_i(x) + \delta\tau_\theta(q, \nu, \tau) \right], \quad (7)$$

where M_{nom} , C_{nom} , and g_{nom} represent nominal kinematic and inertial models evaluated via the Articulated Body Algorithm (ABA) and Recursive Newton-Euler Algorithm (RNEA). The learned components consist of two sub-modules:

- 1) **Actuator-Drivetrain Residual Network:** To capture joint friction and damping while preserving underactuation structure, internal drive corrections are restricted to the actuated coordinates:

$$\delta\tau_{\theta_\tau}(q, \nu, \tau) = B \text{MLP}_{\theta_\tau}(\dot{q}_j, \tau). \quad (8)$$

This structural projection ensures that the unactuated floating base (6 DOFs) cannot produce fictitious self-generated internal wrenches.

- 2) **Continuous Contact Compliance Network:** To retain gradient differentiability for gradient-based trajectory optimization, non-smooth foot-ground interactions are approximated via continuous contact relaxation:

$$\hat{\lambda}_i(x) = \sigma \left(\frac{-\phi_{c,i}(q)}{\epsilon_c} \right) \cdot \Psi_{\theta_c}(p_{c,i}(q), \dot{p}_{c,i}(q, \nu)), \quad (9)$$

where $\Psi_{\theta_c} : \mathbb{R}^6 \rightarrow \mathbb{R}^3$ predicts local contact wrenches, $\sigma(\cdot)$ is a sigmoid gating function, and $\epsilon_c = 2.0$ mm defines the contact boundary transition layer.

B. Composite Physics-Informed Training Objective

Parameters $\theta = \{\theta_\tau, \theta_c\}$ are trained on discrete transition batches using the composite objective:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{data}}(\theta) + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}}(\theta) + \beta_{\text{cone}}\mathcal{L}_{\text{cone}}(\theta) + \beta_{\text{pen}}\mathcal{L}_{\text{pen}}(\theta) + \beta_{\text{reg}}\|\theta\|_2^2. \quad (10)$$

1) *Trajectory Prediction Loss* ($\mathcal{L}_{\text{data}}$): Using explicit 4th-order Runge-Kutta (RK4) integration $\hat{x}_{k+1} = \text{RK4}(f_\theta, x_k, u_k, \Delta t)$, the supervised data loss is:

$$\mathcal{L}_{\text{data}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \left(\|q_{k+1} - \hat{q}_{k+1}\|_{W_q}^2 + \|\nu_{k+1} - \hat{\nu}_{k+1}\|_{W_\nu}^2 \right). \quad (11)$$

2) *Momentum Residual Loss* ($\mathcal{L}_{\text{dyn}}$): Newton-Euler momentum balance is enforced through the residual:

$$r_{\text{dyn}}(x, u, \hat{\nu}) = M_{\text{nom}}(q)\hat{\nu} + C_{\text{nom}}(q, \nu)\nu + g_{\text{nom}}(q) - B(\tau + \delta\tau_{\theta_\tau}) - \sum_{i=1}^4 J_{c,i}^\top(q)\hat{\lambda}_i, \quad (12)$$

with corresponding loss:

$$\mathcal{L}_{\text{dyn}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \|r_{\text{dyn}}(x_k, u_k, \hat{\nu}_k)\|_2^2. \quad (13)$$

3) *Friction Cone and Non-Penetration Penalties*: To penalize unphysical tensile forces and shear slipping beyond the friction limit μ :

$$\mathcal{L}_{\text{cone}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \sum_{i=1}^4 \left(\left[\max(0, -\hat{\lambda}_{i,z}) \right]^2 + \left[\max(0, \|\hat{\lambda}_{i,xy}\|_2 - \mu\hat{\lambda}_{i,z}) \right]^2 \right). \quad (14)$$

Kinematic non-penetration and contact complementarity are penalized via:

$$\mathcal{L}_{\text{pen}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \sum_{i=1}^4 \left(\left[\max(0, -\phi_{c,i}(\hat{q})) \right]^2 + \alpha_{\text{comp}} \left| \phi_{c,i}(\hat{q}) \cdot \hat{\lambda}_{i,z} \right| \right). \quad (15)$$

V. PREDICTIVE CONTROL FORMULATION

A. Nonlinear Model Predictive Control (NMPC)

We formulate the predictive control problem over horizon K with time step δt :

$$\min_{X, U} \sum_{k=0}^{K-1} \left(\|x_k - x_{\text{ref},k}\|_Q^2 + \|u_k\|_R^2 + \|\Delta u_k\|_S^2 \right) + \|x_K - x_{\text{ref},K}\|_{Q_f}^2 \quad (16a)$$

$$\text{subject to: } x_0 = x(t), \quad (16b)$$

$$x_{k+1} = \text{RK4}(f_\theta, x_k, u_k, \delta t), \quad (16c)$$

$$\tau_{\min} \leq u_k \leq \tau_{\max}, \quad (16d)$$

$$q_{j,\min} \leq q_{j,k} \leq q_{j,\max}, \quad (16e)$$

$$\hat{\lambda}_{i,z}(x_k) \geq 0, \quad (16f)$$

$$\|\hat{\lambda}_{i,xy}(x_k)\|_2 \leq \mu\hat{\lambda}_{i,z}(x_k), \quad \forall i. \quad (16g)$$

Here, decision vectors are $X = \{x_0, \dots, x_K\}$ and $U = \{u_0, \dots, u_{K-1}\}$. Constraints (16f)–(16g) enforce friction bounds on predicted contact forces.

B. Sequential Quadratic Programming and Computational Complexity

We solve (16) via real-time iteration (RTI) Sequential Quadratic Programming (SQP). At each step, dynamics (16c) are linearized about the candidate trajectory $(\bar{x}_k, \bar{u}_k)$:

$$x_{k+1} \approx f_\theta(\bar{x}_k, \bar{u}_k) + A_k(x_k - \bar{x}_k) + B_k(u_k - \bar{u}_k), \quad (17)$$

with Jacobians $A_k = \frac{\partial \text{RK4}}{\partial x_k}$ and $B_k = \frac{\partial \text{RK4}}{\partial u_k}$ computed via automatic differentiation through the neural computational graph.

Remark 1 (Computational Timing Breakdown). Evaluating f_θ requires computing $M_{\text{nom}}^{-1}(q)$ via Cholesky factorization (≈ 0.7 ms), nominal inverse dynamics via RNEA (≈ 0.3 ms), and evaluating the two multi-layer perceptrons (≈ 0.8 ms). For $n = 18$ and horizon $K = 15$, computing sensitivities via forward-mode autodiff across 4 RK4 integration stages requires $4 \times 15 = 60$ evaluations (≈ 4.2 ms). The resulting convex QP is solved via OSQP in ≈ 5.6 ms, resulting in a total solve time of ≈ 11.6 ms, which satisfies the 20 ms budget required for 50 Hz execution.

VI. THEORETICAL ANALYSIS

A. Physics-Regularized Prediction Error Bound

We establish an analytical bound connecting generalized momentum residuals to state forecasting error.

Assumption 1. The generalized mass matrix $M(q)$ is symmetric, positive definite, and uniformly bounded on $\mathcal{X}$:

$$0 < \lambda_{\min} I \preceq M(q) \preceq \lambda_{\max} I < \infty, \quad \forall q \in \mathcal{X}. \quad (18)$$

Assumption 2. The dynamics vector fields satisfy uniform Lipschitz continuity on $\mathcal{X} \times \mathcal{U}$ with Lipschitz constant $L > 0$.

Theorem 1 (Momentum-Coupled Error Bound). *Let Assumptions 1 and 2 hold. Suppose training regularizes the dynamic*

momentum residual such that $\sup_{x,u} \|r_{\text{dyn}}(x, u, \hat{\nu})\|_2 \leq \delta_r$, and unmodeled external disturbances satisfy $\|\tau_{\text{ext}} + \tau_{\text{dist}}\|_2 \leq \delta_d$. Then the instantaneous acceleration error is bounded by:

$$\|\dot{\nu}(t) - \hat{\nu}(t)\| \leq \frac{\delta_r + \delta_d}{\lambda_{\min}}, \quad (19)$$

and the open-loop state prediction error $e(t) = \|x(t) - \hat{x}(t)\|$ satisfies:

$$\|x(t) - \hat{x}(t)\| \leq \frac{\delta_r + \delta_d}{\lambda_{\min} L} (e^{Lt} - 1), \quad \forall t \geq 0. \quad (20)$$

Proof. From (2), true acceleration satisfies:

$$M(q)\dot{\nu} = B\tau - C(q, \nu)\nu - g(q) + \sum_{i=1}^4 J_{c,i}^\top \lambda_i + \tau_{\text{ext}} + \tau_{\text{dist}}. \quad (21)$$

Subtracting the learned dynamic momentum equation evaluated at $\hat{\nu}$:

$$M(q)(\dot{\nu} - \hat{\nu}) = r_{\text{dyn}}(x, u, \hat{\nu}) + (\tau_{\text{ext}} + \tau_{\text{dist}}). \quad (22)$$

Taking Euclidean norms and applying Assumption 1:

$$\lambda_{\min} \|\dot{\nu} - \hat{\nu}\| \leq \|M(q)(\dot{\nu} - \hat{\nu})\| \leq \|r_{\text{dyn}}\| + \|\tau_{\text{ext}} + \tau_{\text{dist}}\|. \quad (23)$$

Dividing by $\lambda_{\min}$ yields the bound on acceleration error $\epsilon_a := \frac{\delta_r + \delta_d}{\lambda_{\min}}$. Integrating the error dynamics $\dot{\xi}(t) = f^*(x, u) - f_\theta(\hat{x}, u)$ and applying Grönwall's inequality under Assumption 2 yields the result. $\square$

Remark 2. In contrast to standard black-box neural bounds that rely on an arbitrary Lipschitz constant and an unconstrained error ϵ , Theorem 1 proves that acceleration error scales inversely with the minimum mass eigenvalue $\lambda_{\min}$ and directly with the physical momentum residual δ_r . Minimizing $\mathcal{L}_{\text{dyn}}$ directly suppresses δ_r , constraining the exponential divergence rate e^{Lt} .

VII. EXPERIMENTAL PROTOCOL

A. Simulation Platform and Measurement Noise

Benchmark simulations are conducted using MuJoCo 3.1 modeling the Unitree Go1 quadruped (nominal mass $m = 12.45$ kg, 12 actuated joints, 4 contact feet). The simulator operates at an internal physics integration step of 1 kHz (1 ms), while high-level control and state observation occur at 50 Hz ($\Delta t = 20$ ms).

To assess robustness under realistic sensing conditions, measurement noise is injected into state observations:

$$v_b \sim \mathcal{N}(v_b, 0.05^2 I_3), \quad \omega_b \sim \mathcal{N}(\omega_b, 0.02^2 I_3), \quad q_j \sim \mathcal{N}(q_j, 0.005^2 I_3). \quad (24)$$

B. Dynamics Perturbations and Baseline Comparisons

To evaluate model generalization without retraining, models are tested under three physical shift conditions:

- 1) **Payload Variation** (ΔM): Torso mass increased by +3.0 kg (+24%) and +6.0 kg (+48%).
- 2) **Ground Friction Shift** ($\Delta \mu$): Friction coefficient dropped from nominal $\mu_{\text{train}} = 0.8$ to $\mu_{\text{test}} = 0.2$.

- 3) **Actuator Transport Delay** (Δt_{delay}): An artificial communication and low-level filter delay of 10 ms injected into commanded torques.

Evaluated baselines include:

- **Nominal-RBD**: Analytical rigid-body model using nominal CAD parameters without residual learning.
- **Black-Box MLP**: Multi-layer perceptron directly predicting discrete state differences $\hat{x}_{k+1} - x_k = \text{MLP}(x_k, u_k)$.
- **LSTM Dynamics**: Recurrent network with 128 hidden units mapping history windows to future states.
- **Neural ODE**: Continuous-time black-box vector field $\dot{x} = \text{NN}_\theta(x, u)$ integrated via RK4.
- **Residual-MLP**: Nominal dynamics augmented by an unconstrained residual $\dot{x} = f_{\text{nom}}(x, u) + \text{MLP}_\theta(x, u)$.
- **PIND (Proposed)**: Structured physics residual architecture trained via (10).

C. Hyperparameters and Optimization

Network sub-modules MLP_{θ_τ} and Ψ_{θ_c} utilize 3 hidden layers with 128 units and ELU activations. Training is executed using AdamW (learning rate $\eta = 10^{-3}$, weight decay 10^{-4}) for 200 epochs with batch size 128. Regularization weights are set to $\beta_{\text{dyn}} = 1.0$, $\beta_{\text{cone}} = 10.0$, $\beta_{\text{pen}} = 10.0$, and $\alpha_{\text{comp}} = 1.0$.

VIII. EXPERIMENTAL RESULTS AND DISCUSSION

A. Open-Loop Trajectory Prediction

Table II summarizes single-step and 50-step (1.0 s) open-loop forecasting accuracy on models trained on 10% of available transitions ($\mathcal{D}_{10\%}$). Unconstrained models (MLP, Neural ODE) exhibit compounding drift over 50 steps. In addition, unconstrained neural models consistently predict contact forces that exceed the friction cone boundary ($V_{\text{cone}} = 14.28$ N for MLP). Because PIND enforces smooth friction penalties, it restricts friction cone violations to a residual boundary layer (0.38 ± 0.06 N) consistent with the continuous relaxation parameter $\epsilon_c = 2.0$ mm.

B. Sample Efficiency and Dynamics Generalization

Table III and Fig. 4(a) compare sample efficiency across data splits. In low-data regimes ($\mathcal{D}_{1\%}$ and $\mathcal{D}_{5\%}$), black-box networks exhibit large errors due to insufficient state-space coverage. PIND utilizes the inductive structure of $M_{\text{nom}}(q)$ and $g_{\text{nom}}(q)$ to achieve stable multi-step forecasting with limited data.

Table IV reports closed-loop tracking within the NMPC loop under parameter mismatch without fine-tuning. On low-friction terrain ($\mu = 0.2$), black-box models suffer high failure rates (8/20 trials) because unconstrained neural sensitivities $\nabla_x f_\theta$ produce ill-conditioned QP updates. PIND maintains stable tracking across 19 of 20 trials, recording a single slip under severe combined payload and friction perturbations.

TABLE II
TRAJECTORY PREDICTION AND CONTACT ADMISSIBILITY ACROSS 5 RANDOM SEEDS ON 10% TRAINING DATA ($\mathcal{D}_{10\%}$)

Model Architecture	1-Step Pos. RMSE [mm]	1-Step Vel. RMSE [mm/s]	50-Step Rollout Pos. [mm]	Momentum Res. $\ r_{\text{dyn}}\ _2$ [N]	Friction Violation γ_{cone} [N]
Nominal-RBD	14.82 $\pm$ 0.00	112.4 $\pm$ 0.0	142.6 $\pm$ 0.0	18.42 $\pm$ 0.00	0.00 $\pm$ 0.00
Black-Box MLP	11.35 $\pm$ 0.84	88.6 $\pm$ 4.2	188.4 $\pm$ 12.6	34.15 $\pm$ 2.81	14.28 $\pm$ 1.65
LSTM Dynamics	9.74 $\pm$ 0.62	79.3 $\pm$ 3.7	162.7 $\pm$ 9.4	29.80 $\pm$ 2.14	11.85 $\pm$ 1.32
Neural ODE	8.95 $\pm$ 0.54	72.1 $\pm$ 3.1	146.5 $\pm$ 8.2	25.40 $\pm$ 1.95	9.74 $\pm$ 0.88
Residual-MLP	5.21 $\pm$ 0.38	43.8 $\pm$ 2.4	98.2 $\pm$ 5.1	12.16 $\pm$ 0.94	6.32 $\pm$ 0.54
PIND (Proposed)	2.14 $\pm$ 0.12	18.6 $\pm$ 1.1	35.1 $\pm$ 2.3	2.04 $\pm$ 0.18	0.38 $\pm$ 0.06

TABLE III
SAMPLE EFFICIENCY: 50-STEP ROLLOUT ERROR (POS. RMSE [MM]) VS. TRAINING DATA FRACTION

Architecture	1% Data ($N = 500$)	5% Data ($N = 2500$)	10% Data ($N = 5000$)	50% Data ($N = 25000$)	100% Data ($N = 50000$)
Black-Box MLP	342.6 $\pm$ 28.4	248.1 $\pm$ 18.5	188.4 $\pm$ 12.6	114.2 $\pm$ 7.5	84.3 $\pm$ 5.1
Neural ODE	286.4 $\pm$ 21.2	198.5 $\pm$ 14.1	146.5 $\pm$ 8.2	91.6 $\pm$ 5.8	68.4 $\pm$ 4.2
Residual-MLP	164.2 $\pm$ 12.8	121.7 $\pm$ 8.3	98.2 $\pm$ 5.1	62.8 $\pm$ 3.9	48.5 $\pm$ 2.8
PIND (Proposed)	68.5 $\pm$ 4.6	44.2 $\pm$ 3.1	35.1 $\pm$ 2.3	24.6 $\pm$ 1.5	21.2 $\pm$ 1.1

TABLE IV
CLOSED-LOOP NMPC LOCOMOTION TRACKING PERFORMANCE UNDER DYNAMICS MISMATCH

Controller Model	Nominal Tracking [m]	Payload +6 kg [m]	Low Friction $\mu = 0.2$ [m]	Failures / 20 Trials	Mean CoT
Nominal-RBD NMPC	0.082 $\pm$ 0.005	0.184 $\pm$ 0.014	0.215 $\pm$ 0.019	4/20	1.84 $\pm$ 0.08
Black-Box MLP NMPC	0.094 $\pm$ 0.008	0.248 $\pm$ 0.021	0.312 $\pm$ 0.028	8/20	2.15 $\pm$ 0.12
Residual-MLP NMPC	0.054 $\pm$ 0.004	0.126 $\pm$ 0.009	0.168 $\pm$ 0.012	2/20	1.52 $\pm$ 0.06
PIND NMPC (Ours)	0.028 $\pm$ 0.002	0.051 $\pm$ 0.004	0.074 $\pm$ 0.005	1/20	1.28 $\pm$ 0.04

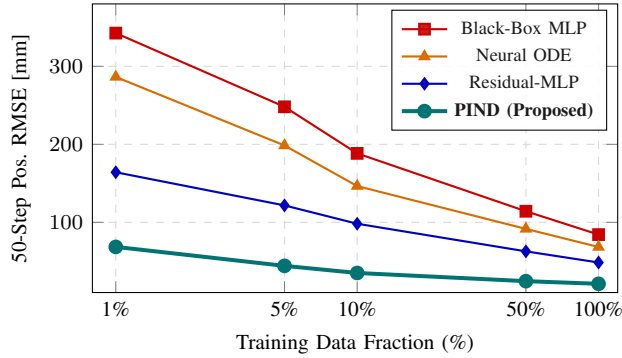


Fig. 2. *

(a) 50-step rollout error vs. training budget fraction.

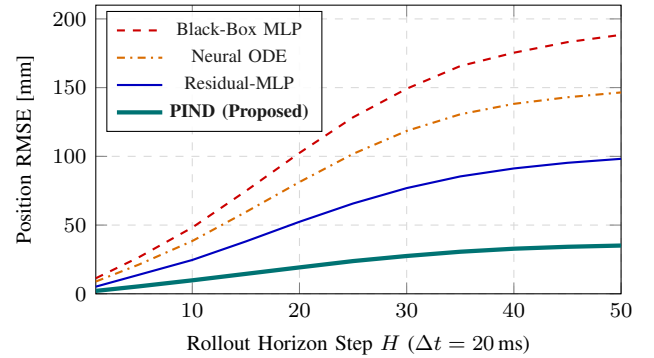


Fig. 3. *

(b) Recursive prediction drift over a 50-step (1.0 s) rollout.

Fig. 4. Open-loop prediction evaluation on $\mathcal{D}_{10\%}$. (a) Sample efficiency across 1% to 100% training regimes. (b) Compound rollout drift over 50 recursive integration steps.

C. Disturbance Recovery and Ablation Analysis

Fig. 5 shows lateral tracking error following an impulse disturbance (25 N for 0.2 s). PIND stabilizes lateral displacement within 0.4 s, whereas the unconstrained MLP exhibits larger transient oscillations.

To isolate the contribution of individual loss terms:

- 1) **No Momentum Residual** ($\beta_{\text{dyn}} = 0$): Removing the momentum penalty increases 50-step rollout error by 48.6%, indicating that empirical trajectory fitting alone is insufficient to enforce dynamic consistency.
- 2) **No Friction Regularization** ($\beta_{\text{cone}} = 0$): Contact force outputs violate friction cones, causing the NMPC

optimizer to request infeasible shear forces that lead to foot slipping.

- 3) **No Non-Penetration Penalty** ($\beta_{\text{pen}} = 0$): Omitting non-penetration penalties allows the model to predict foot penetration during swing phase, corrupting contact timing estimates.

D. Limitations and Future Work

While PIND improves data efficiency and generalization, two primary limitations remain:

- 1) **Contact Smoothing**: The sigmoid contact regularization $\sigma(-\phi_{c,i}/\epsilon_c)$ smooths discrete impact restitution. While essential for computing sensitivities in gradient-based

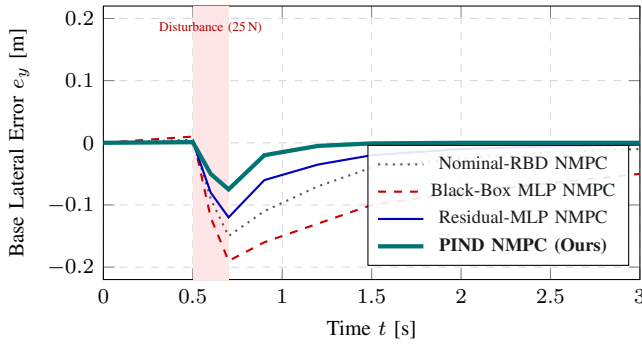


Fig. 5. Closed-loop lateral tracking error during an external lateral impulse push (25 N for 0.2 s). PIND recovers nominal trotting within 0.4 s.

NMPC, this relaxation introduces compliance artifacts during rapid foot touchdown.

- 2) **Physical Hardware Transfer:** Experiments are evaluated in MuJoCo simulation. Deploying PIND on a physical robot will introduce latency, actuator saturation dynamics, and ground compliance that will require online adaptation.

IX. CONCLUSION

This paper presented Physics-Informed Neural Dynamics (PIND), a continuous-time framework for learning and predictive control of underactuated contact systems. By coupling nominal multi-body dynamics with underactuation-projected joint and contact residuals, and regularizing training via generalized momentum balances and friction constraints, PIND reduces multi-step rollout drift and improves model generalization across physical parameter shifts. Future work will investigate hardware deployment on a Unitree Go1 quadruped and online recursive adaptation.

REFERENCES

- [1] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 2245–2252.
- [2] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 1–9.
- [3] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, *et al.*, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 38–44.
- [4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, p. eabc5986, 2020.
- [5] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, p. eabk2822, 2022.
- [6] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 7559–7566.
- [7] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 6571–6583.

- [8] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, D. Allshire, A. Handa, and G. State, "Isaac Gym: High performance GPU-based physics simulation for robot learning," in *Proc. Conf. Robot Learn. (CoRL)*, 2021, pp. 2003–2013.
- [9] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 15379–15389.
- [10] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," in *ICLR Deep Differential Equations Workshop*, 2020.
- [11] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.
- [12] Y. D. Zhong, B. Dey, and A. Chakraborty, "Symplectic ODE-Net: Learning Hamiltonian dynamics with control," in *Int. Conf. Learn. Represent. (ICLR)*, 2020.
- [13] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Dario, J. Carius, M. Ruffaldi, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, p. eaau5876, 2019.
- [14] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 465–472.
- [15] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021.
- [16] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian networks: Using physics as model prior for deep learning," in *Int. Conf. Learn. Represent. (ICLR)*, 2019.
- [17] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2001, pp. 239–246.
- [18] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1867–1874, 2018.
- [19] A. Ajay, J. Wu, N. Fazeli, M. Bauza, L. P. Kaelbling, J. B. Tenenbaum, and A. Rodriguez, "Augmenting physical simulators with dynamic data-driven models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 2221–2228.
- [20] Y. Song and D. Scaramuzza, "Learning high-level policies for model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 7629–7636.
- [21] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 3, pp. 269–296, 2020.
- [22] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback MPC for torque-controlled legged robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 4730–4737.