2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning Forward Looking Adaptation to Dynamic Payloads for Quadruped Locomotion via Physics-Informed Neural Networks

Oscar Youngquist and Hao Zhang

***Abstract*— Payload-adaptive locomotion is an essential capability for quadruped robots operating in real-world scenarios, particularly when tasked with transporting dynamic payloads. Existing approaches face fundamental limitations: reactive adaptation strategies respond too slowly to sudden payload changes, while learning-based methods often yield physically inconsistent models of robot dynamics that generalize poorly to novel states. To address these key challenges, we introduce Forward-Looking Adaptation to Dynamic Payloads (FLAP), a novel approach that learns to proactively compensate for discrepancies between expected and actual locomotion behavior induced by dynamic payloads. FLAP combines two critical components: (1) a physics-informed neural network (PINN) that predicts anticipated joint states while enforcing physical consistency through dynamics based loss functions, and (2) a composite adaptive control law that rapidly generates anticipatory joint torque compensations based on the PINN's predictions. Through unifying structured dynamics modeling with real-time anticipatory control, our method enables generalizable and physically consistent adaptation to dynamic payloads. Experimental results demonstrate that FLAP achieves robust locomotion under diverse payload conditions on physical quadruped robots in real-world environments. More details available on the project website: `https://hcrlab.gitlab.io/project/flap`.**

I. Introduction

Quadruped robots have demonstrated remarkable advances in mobility driven by their potential to operate in unstructured environments [1]–[5]. These capabilities have enabled their use in applications such as autonomous inspection [6]–[9], disaster response [10], [11], and search-and-rescue [12], [13]. In such applications, legged robots may be tasked with transporting dynamic payloads, including shifting rubble, loosely packed containers, or liquids (Fig. 1). These dynamic payloads induce time-varying mass distributions that generate additional forces and torques acting on the robot's body. To maintain stable locomotion, the quadruped robot must rapidly adapt to these disturbances by compensating for the resulting deviations in real time.

The importance and challenges of payload-adaptive legged locomotion have motivated the development of a wide range of approaches. Model-based approaches leverage control-theoretic principles and explicit knowledge of system dynamics [14]–[17], but they often adapt slowly to abrupt payload changes and require significant manual tuning. Learning-based approaches [2], [3], [18]–[21] have demonstrated promising results, but current state-of-the-art methods frequently fail to learn physically-consistent representations of the robot's dynamics. This limitation results in significant performance degradation when the robot encounters novel states outside the training distribution, which underscores a critical gap in robust generalization for real-world deployment under dynamic payloads.

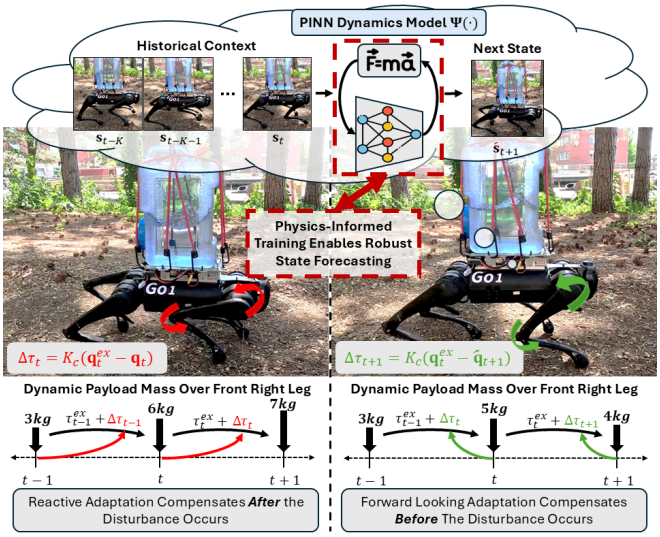


Fig. 1. Transporting dynamic payloads (e.g., liquids) is challenging due to time-varying mass distributions that overwhelm reactive control policies. Our physics-informed robot learning method enables accurate state forecasting for "forward looking" proactive adaptation, which achieves robust real-world transport of dynamic payloads.

We introduce the *Forward Looking Adaptation to Dynamic Payloads* (FLAP) approach (Fig. 1) for real-time adaptation of quadruped locomotion controllers to dynamic payloads. FLAP enables proactive compensation for anticipated locomotion discrepancies caused by shifting payload mass distributions through a physics-informed neural network (PINN) that is designed to model the robotic system's forward dynamics. The PINN simultaneously predicts the evolution of the robot's whole body state and the payload-induced forces and torques acting on the robot, while a physics-informed loss function ensures the predictions respect the dynamics of the quadruped. Through PINN, FLAP is able to learn a physically plausible forward dynamics model that generalizes well to novel states. The learned dynamics are then integrated into a composite adaptive control law that enables rapid, anticipatory joint-torque compensation in real time during execution. Through integrating physics-informed learning with adaptive control, FLAP enables robust quadruped locomotion with effective

*This work was partially supported by NSF CAREER Award IIS-2308492, DARPA Young Faculty Award (YFA) D21AP10114-00, NSF award IIS-2404386, and DARPA award N652362690001.

Oscar Youngquist and Hao Zhang are with Human-Centered Robotics Lab, University of Massachusetts Amherst, Amherst, MA 01002, USA. Email: {oyoungquist, hao.zhang}@umass.edu.

adaptation to dynamic payloads.

The main contribution of this paper is the introduction of FLAP, a novel physics-informed learning approach to address dynamic payload adaptation in quadruped robots. Two specific novelties of our work include:

- We introduce a novel physics-informed neural network for modeling physically consistent robot dynamics, enabling improved generalization to previously unseen scenarios during deployment.
- We introduce a new capability for rapid forward-looking proactive adaptation to changing mass distributions of dynamic payloads, enhancing quadruped robot's ability to maintain stable and efficient locomotion under varying dynamic load conditions in real-world payload transportation tasks.

II. Related Work

A. Non-Learning Approaches for Payload Adaptation

Non-learning approaches to adaptive quadruped locomotion typically rely on analytical dynamics modeling and control-theoretic frameworks. Early methods employed online payload identification techniques [14], [22] or force-based quadratic programming [15]. Subsequent advances introduced stability-guaranteed formulations using Control Lyapunov Functions (CLFs) [16] to ensure energy dissipation toward desired states. The method proposed in [17] integrates L1 adaptive control techniques [23], consisting of a fast adaptation law and a low-pass filter to decouple estimation and control, into a force-based model predictive control (MPC) method. Recently, [22] proposes a gradient-descent-based adaptive control law leveraging the low-dimensional representations for non-linear and complex locomotion models known as templates [24]. While theoretically principled, these methods often require complex control architectures with substantial computational overhead and significant domain expertise to implement effectively.

B. Learning Approaches for Payload Adaptation

Current approaches for learning adaptive quadruped locomotion employ reinforcement learning (RL) with varying adaptation strategies. Privileged learning methods [3], [18], [19] leverage "privileged" state data (e.g., payload mass, terrain properties) only available in simulation to learn adaptive policies. The adaptive ability of these approaches comes from large amounts of training data across a wide range of procedurally generated conditions. However their performance is limited to conditions encountered during training [25]. Online learning techniques [20], [21] address this limitation through in-situ training, though their slow convergence limits responsiveness to sudden payload changes. Hybrid approaches combine RL with classical adaptive control, either by modifying MPC inputs/outputs [26], [27] or through complementary adaptive controllers [28], [29], but remain constrained to learned configurations. Recent work explores hierarchical policies [30], [31], that first learn a nominal locomotion policy followed by an complimentary adaptive policy for generating compensation values. However, these methods all share a critical limitation: they neglect the underlying differential equations governing robot dynamics, resulting in physically inconsistent models that impair generalization.

C. Physics Informed Neural Networks (PINNs)

Physics-informed neural networks (PINNs) model dynamical systems by embedding their governing differential equations directly into the training objective, enforcing physical consistency via equation-based loss terms leveraging automatic differentiation tools [32]. While widely applied in scientific computing problems such as fluid dynamics [33], [34], heat transfer [35], and solid mechanics [36], [37], PINNs have recently gained traction in robotics [38]. Applications include modeling manipulator dynamics [39], quadrotor dynamics [40], [41], soft robots [42], [43], state estimation [44], [45], and motion planning [46], [47], demonstrating their potential for learning physically accurate representations of robotic systems.

III. Approach

Notation: Matrices are denoted as boldface uppercase letters (e.g. $\mathbf{M} = \{M_{i,j}\} \in \mathbb{R}^{n\times m}$ is an $n \times m$ matrix whose element in the i-th row and j-th column is $M_{i,j}$), vectors are denoted as boldface lowercase letters (e.g., $\mathbf{v} \in \mathbb{R}^d$ is a d-dimensional vector whose i-th element is v_i), and scalars are represented by non-boldface characters (e.g., s, α).

A. Forward Looking Dynamic Payload Compensation Problem Formulation

During locomotion, a quadruped robot carrying a payload gathers proprioceptive observations through its onboard sensors. Joint encoders provide leg positions $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$, while an IMU measures torso orientation Θ, angular velocity $\dot{\Theta}$, and linear acceleration $\ddot{\mathbf{p}}$. A Kalman filter estimates the global torso position $\mathbf{p}$ and velocity $\dot{\mathbf{p}}$ from kinematic data and IMU measurements. The system also computes exerted joint torques $\boldsymbol{\tau}$ using $\dot{\mathbf{q}}$, motor current, and models of the joint actuators. At each timestep, proprioceptive observations are aggregated into a state vector $\mathbf{s} = [\dot{\mathbf{p}}, \Theta, \dot{\Theta}, \mathbf{q}, \dot{\mathbf{q}}]$, which serves as input to the locomotion controller.

The expected locomotion behavior is defined by expected torso velocities $\mathbf{b}^{ex} = [\dot{\mathbf{p}}^{ex}, \dot{\Theta}^{ex}]^\top$, typically provided by a motion planner or user input. A parameterized controller $\Phi(\cdot)$ then uses $\mathbf{s}$ and $\mathbf{b}^{ex}$ to generate joint position and torque commands $\Phi : (\mathbf{s}, \mathbf{b}^{ex}) \mapsto (\mathbf{q}^{ex}, \boldsymbol{\tau}^{ex})$. $\Phi(\cdot)$ encodes the quadruped robot's dynamics which can be expressed as:

$$\begin{pmatrix} \mathbf{0}_6 \\ \boldsymbol{\tau} \end{pmatrix} + \mathbf{J}_c^\top \mathbf{f}^r = \mathbf{A}\ddot{\boldsymbol{\rho}} + \mathbf{c} + \mathbf{g} \tag{1}$$

where $\ddot{\boldsymbol{\rho}} = [\ddot{\mathbf{p}}, \ddot{\Theta}, \ddot{\mathbf{q}}]^\top$, $\mathbf{A}$ (generalized mass matrix), $\mathbf{c}$ (Coriolis forces), $\mathbf{g}$ (gravitational forces), $\mathbf{J}_c^\top$ (contact Jacobian), and $\mathbf{f}^r$ (ground reaction forces) characterize the system dynamics. The null vector $\mathbf{0}_6$ corresponds to the torso's unactuated degrees of freedom. Eq. (1) represents the second-order dynamics relating robot accelerations to applied forces/torques.

When transporting a payload, the additional mass shifts the system dynamics away from Eq. (1). This manifests primarily as joint torque disturbances $\Delta\boldsymbol{\tau} = \mathbf{J}_c^\top \Delta\mathbf{f}^r$, where $\Delta\mathbf{f}^r$ represents unmodeled variations in ground reaction forces caused by the mass of the payload. The modified dynamics now take the form:

$$\begin{pmatrix}\mathbf{0}_6\\ \boldsymbol{\tau}\end{pmatrix} + \mathbf{J}_c^\top \mathbf{f}^r + \Delta\boldsymbol{\tau} = \mathbf{A}\ddot{\boldsymbol{\rho}} + \mathbf{c} + \mathbf{g} \tag{2}$$

This model mismatch causes $\mathbf{q}$ to deviate from $\mathbf{q}^{ex}$, resulting in position errors $\Delta\mathbf{q} = \mathbf{q}^{ex} - \mathbf{q} \neq \mathbf{0}$ that degrade locomotion performance.

In order to account for the $\Delta\boldsymbol{\tau}$ introduced by the payload, we compensate $\Phi(\cdot)$ with joint torque offsets:

$$\boldsymbol{\tau}^{ex} = \Phi(\mathbf{s}, \mathbf{b}^{ex}) + \Delta\boldsymbol{\tau} \tag{3}$$

While direct measurement of $\Delta\boldsymbol{\tau}$ is infeasible without foot force sensors, we leverage adaptive control principles [28], [48], [49] to estimate it via joint position errors:

$$\Delta\boldsymbol{\tau} \approx \mathbf{K}_c \Delta\mathbf{q} = \mathbf{K}_c(\mathbf{q}^{ex} - \mathbf{q}) \tag{4}$$

where $\mathbf{K}_c$ is an adaptive gain matrix. As a result, the compensated torque command explicitly accounts for unmodeled dynamics through measurable state deviations.

This formulation follows a "reactive" adaptive control paradigm, where state feedback compensates for disturbances in real time. However, reactive control lacks foresight: the commanded joint state $\mathbf{q}^{ex}$ targets the next timestep, while $\mathbf{q}$ reflects only the current state. Since compensation via $\Delta\mathbf{q}$ relies solely on instantaneous sensory inputs, it cannot preemptively counteract impending disturbances from a dynamic payload. To address this, we propose a proactive approach using a learned forward dynamics model $\Omega(\cdot)$. By processing a history of K expected behavior, state, and previous-action triplets $\mathbf{X}_t = [\mathbf{x}_{t-K}, \ldots, \mathbf{x}_t]^\top$, where $\mathbf{x}_i = [\mathbf{b}_i^{ex}, \mathbf{s}_i, \mathbf{q}_{i-1}^{ex}]^\top$, $\Omega(\cdot)$ predicts the future joint state $\hat{\mathbf{q}}_{t+1} = \Omega(\mathbf{X}_t)$. This enables anticipatory compensation via:

$$\Delta\boldsymbol{\tau}_{t+1} = \mathbf{K}_c(\mathbf{q}_t^{ex} - \hat{\mathbf{q}}_{t+1}) \tag{5}$$

effectively transforming the adaptation procedure from reactive to proactive, enabling "forward looking" adaptive locomotion for dynamic payloads.

The problem of learning $\Omega(\cdot)$ can be formulated as:

$$\underset{\Omega}{\operatorname{argmin}} \sum_{(\mathbf{X}_i, \mathbf{q}_{i+1}) \in \mathbf{D}} \|\mathbf{q}_{i+1} - \Omega(\mathbf{X}_i)\|^2 \tag{6}$$

where $\mathbf{D}$ contains N samples across varied payloads and locomotion behaviors. This standard regression formulation trains $\Omega(\cdot)$ to minimize the next-step joint prediction error to enable proactive compensation as in Eq. (5). However, two key limitations arise: (1) the physics-agnostic approach lacks explicit incorporation of the robot's physical dynamics [25], resulting in (2) exhaustive training data requirements to avoid overfitting [3], [18], [19]. Consequently, without sufficient coverage, the model generalizes poorly to unseen states during deployment.

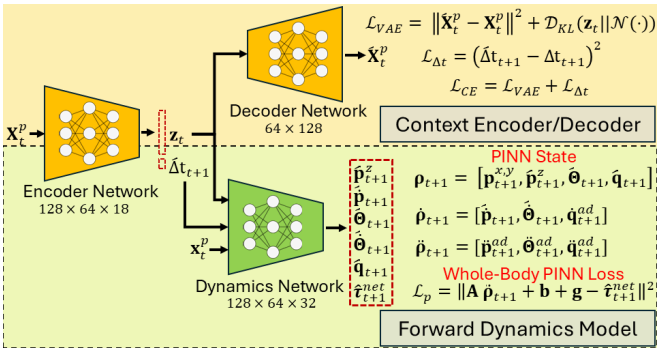


Fig. 2. **PINN Architecture**: The context encoder processes historical data to produce a latent representation and timestep prediction, while the forward dynamics model uses these outputs to predict future states and forces/torques via physics informed learning.

B. PINN for Learning Forward Looking Adaptation to Dynamic Payloads

We present a novel physics informed neural network approach for learning FLAP while addressing the aforementioned limitations. Our approach trains a model $\Psi : \mathbf{X}_t \mapsto (\bar{\mathbf{s}}_{t+1}, \boldsymbol{\tau}_{t+1}^{net})$ to jointly predict the robot's future state and the net forces/torques acting on the robot. The state prediction $\bar{\mathbf{s}}_{t+1} = [\mathbf{p}_{t+1}^z, \Theta_{t+1}, \mathbf{q}_{t+1}]^\top$ captures torso height, orientation, and joint positions (excluding torso x, y position for generalizability), while the force/torque prediction $\boldsymbol{\tau}_t^{net} = [\mathbf{f}_{t+1}^{torso}, \boldsymbol{\tau}_{t+1}^{torso}, \boldsymbol{\tau}_{t+1}]^\top + \mathbf{J}_c^\top \mathbf{f}_{t+1}^r$ includes torso forces $\mathbf{f}^{torso}$, torques $\boldsymbol{\tau}^{torso}$, and payload induced ground reaction forces $\mathbf{J}_c^\top \mathbf{f}^r$ (available as privileged simulation data). Through this formulation, $\Psi(\cdot)$ explicitly learns to model ground reaction force disturbances ($\mathbf{J}_c^\top \Delta\mathbf{f}^r$) while capturing the coupled robot-payload dynamics governing state evolution.

While jointly predicting $\bar{\mathbf{s}}_{t+1}$ and $\boldsymbol{\tau}_{t+1}^{net}$ captures coupled dynamics, this naive approach lacks an explicit connection to the robot's governing dynamics, risking overfitting to $\mathbf{D}$ without learning a physically consistent representation of the robot's dynamics. We address this by formulating $\Psi(\cdot)$ as a Physics Informed Neural Network (PINN) [32], decomposed into two jointly-trained components as seen in Fig. 2. A context encoder/decoder (Ψ_{ENC}/Ψ_{DEC}) processes historical system states and a forward dynamics model (Ψ_{FD}) predicts future states and forces

The context encoder, $\Psi_{ENC} : \mathbf{X}_t^p \mapsto (\mathbf{z}_t, \hat{\Delta}t_{t+1})$, processes a time-augmented history $\mathbf{X}_t^p = [\mathbf{x}_{t-K}^p, \ldots, \mathbf{x}_t^p]^\top$, where each $\mathbf{x}_i^p = [\Delta t_i, \mathbf{b}_i^{ex}, \mathbf{s}_i, \mathbf{q}_{i-1}^{ex}]^\top$ contains the timestep Δt_i between action $\mathbf{q}_{i-1}^{ex}$ and resulting state $\mathbf{s}_i$. The encoder outputs a latent dynamics representation $\mathbf{z}_t$ and predicts the next timestep $\hat{\Delta}t_{t+1}$, while a decoder, $\Psi_{DEC} : \mathbf{z}_t \mapsto \hat{\mathbf{X}}_t^p$, reconstructs $\mathbf{X}_t^p$ from $\mathbf{z}_t$ to ensure it preserves essential features. The forward dynamics model, $\Psi_{FD} : (\mathbf{z}_t, \hat{\Delta}t_t, \mathbf{x}_t^p) \mapsto (\bar{\mathbf{s}}_{t+1}, \boldsymbol{\tau}_{t+1}^{net})$ then uses $\mathbf{z}_t$, $\hat{\Delta}t_{t+1}$, and $\mathbf{x}_t^p$ to predict the next state $\bar{\mathbf{s}}_{t+1}$ and net forces/torques $\boldsymbol{\tau}_{t+1}^{net}$, explicitly modeling the time-dependent system evolution in Eq. (2).

We encourage $\Psi(\cdot)$ to learn a physically consistent representation of the robot's dynamics by constructing loss functions that penalize learning inaccurate higher-order behaviors (e.g. system velocities and accelerations). The higher-order

representations learned by $\Psi(\cdot)$ can be derived using the automatic differentiation (AD) tools implemented in deep learning programming frameworks. For joint dynamics, we calculate $\Psi(\cdot)$'s learned relationship between position and velocity as $\dot{\mathbf{q}}^{ad}_{t+1} = f_{ad}(\hat{\Delta}t_{t+1}, \hat{\mathbf{q}}_{t+1})$ via AD and minimize:

$$\mathcal{L}^{p}_{\dot{\mathbf{q}}_{t+1}} = \|\dot{\mathbf{q}}_{t+1} - \dot{\mathbf{q}}^{ad}_{t+1}\|^2 \tag{7}$$

where $f_{ad}(x, y) = \frac{\partial}{\partial x} y$. $\mathcal{L}^{p}_{\dot{\mathbf{q}}_{t+1}}$ explicitly encourages $\Psi(\cdot)$ to learn a physically consistent relationship between the position and velocity of the next timestep's leg joints that aligns with the system's known dynamics. Due to the x, y position of the torso being excluded from the learning objective, we directly predict $\dot{\mathbf{p}}_{t+1}$ and also $\dot{\Theta}_{t+1}$ due to their tight coupling, yielding the complete state prediction target $\mathbf{s}^{p}_{t+1} = [\mathbf{p}^{z}_{t+1}, \Theta_{t+1}, \mathbf{q}_{t+1}, \dot{\mathbf{p}}_{t+1}, \dot{\Theta}_{t+1}]^\top$.

To encourage $\Psi(\cdot)$'s consistency with the dynamics in Eq. (2), we further compute second-order derivatives via AD. This results in joint accelerations $\ddot{\mathbf{q}}^{ad} = f_{ad}(\hat{\Delta}t, \hat{\dot{\mathbf{q}}}^{ad})$, and torso linear/angular accelerations $\ddot{\mathbf{p}}^{ad}_{t+1} = f_{ad}(\hat{\Delta}t, \hat{\dot{\mathbf{p}}}_{t+1})$, $\ddot{\Theta}^{ad}_{t+1} = f_{ad}(\hat{\Delta}t, \hat{\dot{\Theta}}_{t+1})$. Using the predicted state and approximated higher-order derivatives, the whole body state of the system used by the PINN can be constructed as follows. The position is defined as $\boldsymbol{\rho}^{p}_{t+1} = [\mathbf{p}^{x}_{t+1}, \mathbf{p}^{y}_{t+1}, \hat{\mathbf{p}}^{z}_{t+1}, \hat{\Theta}_{t+1}, \hat{\mathbf{q}}_{t+1}]^\top$ (with the torso's x, y position copied from ground truth), with the velocities being $\dot{\boldsymbol{\rho}}^{p}_{t+1} = [\hat{\dot{\mathbf{p}}}_{t+1}, \hat{\dot{\Theta}}_{t+1}, \dot{\mathbf{q}}^{ad}_{t+1}]^\top$, and accelerations defined as $\ddot{\boldsymbol{\rho}}^{p}_{t+1} = [\ddot{\mathbf{p}}^{ad}_{t+1}, \ddot{\Theta}^{ad}_{t+1}, \ddot{\mathbf{q}}^{ad}_{t+1}]^\top$. The PINN state is then used to encourage learned consistency with the whole-body dynamics in Eq. (2) via the loss function:

$$\mathcal{L}^{p}_{wb} = \|\mathbf{A}\ddot{\boldsymbol{\rho}}^{p}_{t+1} + \mathbf{b} + \mathbf{g} - \hat{\boldsymbol{\tau}}^{net}_{t+1}\|^2 \tag{8}$$

where $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{g}$ are calculated using $\boldsymbol{\rho}^{p}_{t+1}$ and $\dot{\boldsymbol{\rho}}^{p}_{t+1}$. $\mathcal{L}^{p}_{wb}$ explicitly couples state predictions ($\hat{\mathbf{s}}^{p}_{t+1}$) and force/torque estimates ($\hat{\boldsymbol{\tau}}^{net}_{t+1}$) through the system dynamics, ensuring physical consistency.

Putting all these elements together we arrive in the final physics informed loss function:

$$\mathcal{L}_{\text{PINN}} = \mathcal{L}_{CE} + \mathcal{L}^{p}_{n} + \mathcal{L}^{p}_{wb} + \mathcal{L}^{p}_{\dot{\mathbf{q}}_{t+1}} \tag{9}$$

$\mathcal{L}_{\text{PINN}}$ combines four key loss components: the context encoder-decoder loss $\mathcal{L}_{CE}$, the next-state prediction loss $\mathcal{L}^{p}_{n}$, the whole-body dynamics loss $\mathcal{L}^{p}_{wb}$, and the joint velocity loss $\mathcal{L}^{p}_{\dot{\mathbf{q}}_{t+1}}$. The context loss $\mathcal{L}_{CE} = \|\mathbf{X}^{p}_{t} - \hat{\mathbf{X}}^{p}_{t}\|^2 + D_{KL}(\mathbf{z}_t|\mathcal{N}(0,1)) + (\Delta t_{t+1} - \hat{\Delta}t_{t+1})^2$ simultaneously optimizes for accurate sequence reconstruction, latent space regularization, and timestep prediction. The prediction loss $\mathcal{L}^{p}_{n} = \|\mathbf{s}^{p}_{t+1} - \hat{\mathbf{s}}^{p}_{t+1}\|^2 + \|\boldsymbol{\tau}^{net}_{t+1} - \hat{\boldsymbol{\tau}}^{net}_{t+1}\|^2$ ensures precise forecasting of both the robot's state and net forces/torques. Physical consistency is learned through $\mathcal{L}^{p}_{wb}$, defined in Eq. (8), which incorporates the full system dynamics, along with $\mathcal{L}^{p}_{\dot{\mathbf{q}}_{t+1}}$, from Eq. (7), that specifically maintains proper joint velocity relationships. Together, these loss terms enable $\Psi(\cdot)$ to learn physically consistent predictions while effectively summarizing historical observations. Complete implementation details of the training procedure are available in the supplementary material.

Algorithm 1: FLAP Online Execution

Input: $\Psi(\cdot), \mathbf{s}_t, \mathbf{b}^{ex}_t, \mathbf{X}^{p}_t, \kappa, k_0, \lambda_0$
Output: $\mathbf{q}^{ex}_t, \boldsymbol{\tau}^{ex}_t$

while ***running*** **do**
  $\mathbf{q}^{ex}_t, \boldsymbol{\tau}^{ex}_t \leftarrow \Phi(\mathbf{s}_t, \mathbf{b}^{ex}_t)$, $\hat{\mathbf{q}}_{t+1} \leftarrow \Psi(\mathbf{X}^{p}_t)$
  repeat
    get feedback $\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t$
    update $\boldsymbol{\epsilon}_{\mathbf{q}}$ and $\boldsymbol{\epsilon}^{\top}_{\tau}$, update $\lambda(t)$ with Eq. (13)
    update $\Gamma(t)$ with Eq. (12), update $\mathbf{K}_c$ with Eq. (11)
    $\Delta\boldsymbol{\tau}_{t+1} \leftarrow \mathbf{K}_c(\mathbf{q}^{ex}_t - \hat{\mathbf{q}}_{t+1})$
    return $\boldsymbol{\tau}^{ex}_t + \Delta\boldsymbol{\tau}_{t+1}$
  until *next* $\mathbf{q}^{ex}_t$,$\boldsymbol{\tau}^{ex}_t$, *and* $\hat{\mathbf{q}}_{t+1}$
  update $\mathbf{s}$, $\mathbf{b}^{ex}$, $\mathbf{X}^{p}$

C. Online Forward Looking Adaptive Control

The trained model $\Psi(\cdot)$ is incorporated into the adaptive control framework in Eq. (3) through:

$$\boldsymbol{\tau}^{ex}_t = \Phi(\mathbf{s}_t, \mathbf{b}^{ex}_t) + \mathbf{K}_c(\mathbf{q}^{ex}_t - \Psi(\mathbf{X}^{p}_t)) \tag{10}$$

Here, $\Phi(\cdot)$ generates nominal joint commands ($\mathbf{q}^{ex}_t$) and torques ($\boldsymbol{\tau}^{ex}_t$), while $\Psi(\cdot)$ provides physics-informed state predictions $\hat{\mathbf{q}}_{t+1}$. The gain matrix $\mathbf{K}_c$ transforms anticipated state discrepancies into compensatory torques, effectively approximating the required disturbance compensation $\Delta\boldsymbol{\tau}_{t+1}$.

During deployment, the values of $\mathbf{K}_c$ are updated in order to minimize the forward looking tracking errors $\Delta\mathbf{q}_{t+1} = \mathbf{q}^{ex}_t - \hat{\mathbf{q}}_{t+1}$ through adaptive control techniques. A bounded-gain-forgetting (BGF) composite adaptive control law [28], [48], [49] is developed:

$$\dot{\mathbf{K}}_c = -\Gamma(t)\Delta\mathbf{q}_{t+1}(\boldsymbol{\epsilon}^{\top}_{\mathbf{q}} + \kappa\boldsymbol{\epsilon}^{\top}_{\tau}) \tag{11}$$

where $\boldsymbol{\epsilon}_{\mathbf{q}} = \dot{\mathbf{q}}_t - \alpha(\mathbf{q}^{ex}_t - \mathbf{q}_t)$ and $\boldsymbol{\epsilon}^{\top}_{\tau} = \boldsymbol{\tau}^{ex}_t - \boldsymbol{\tau}_t$ represent the joint position and torque tracking errors, respectively, and κ controls the contribution of $\boldsymbol{\epsilon}^{\top}_{\tau}$. $\Gamma(t)$ represents a time dependent positive-definite update matrix which evolves according to:

$$\frac{d}{dt}\Gamma(t) = \lambda(t)\Gamma(t) - \Gamma(t)\Delta\mathbf{q}_{t+1}\Delta\mathbf{q}^{\top}_{t+1}\Gamma(t) \tag{12}$$

with $\lambda(t)$ defined as:

$$\lambda(t) = \lambda_0\left(1 - \frac{\|\Gamma(t)\|}{k_0}\right) \tag{13}$$

λ_0 and k_0 are constants defining the upper bound of $\lambda(t)$ and $\|\Gamma(t)\|$ respectively, ensuring bounded gains. Prior work [48], [49] shows that BGF adaptive update law achieves global asymptotic stability with $\boldsymbol{\epsilon}_{\mathbf{q}}$ and $\boldsymbol{\epsilon}^{\top}_{\tau}$ asymptotically converging to zero.

Algorithm 1 implements our FLAP approach through two parallel processes running at different frequencies: an outer loop (0.5 kHz) generating baseline commands via $\Phi(\cdot)$ and $\Psi(\cdot)$, and an inner loop (1 kHz) executing the BGF adaptation: Eqs. (10)-(13). The outer loop computes $\mathbf{q}^{ex}_t$, $\boldsymbol{\tau}^{ex}_t$, and $\hat{\mathbf{q}}_{t+1}$ while updating the models inputs $\mathbf{s}_t, \mathbf{b}^{ex}_t, \mathbf{X}^{p}_t$, while the inner loop applies the BGF update law using the predicted values to produce $\Delta\boldsymbol{\tau}_{t+1}$. This multi-rate architecture enables rapdily generating forward looking joint torque compensation values for rapidly adapting $\Phi(\cdot)$ to dynamic payloads.

TABLE I

RMSE QUANTITATIVE RESULTS OF FLAP AND COMPARISON METHODS FOR SUCCESSFUL TRAILS ACROSS MULTIPLE PAYLOAD CONDITIONS AND MOTIONS IN SIMULATION. BEST PERFORMANCE IS **BOLDED** AND SECOND BEST IS UNDERLINED.

Condition	Failure Rate (/40)				Torso-Position (m)				Torso-Orientation (rad)				Joint-Position (rad)			
	WBIC	REACT	F-NP	FLAP	WBIC	REACT	F-NP	FLAP	WBIC	REACT	F-NP	FLAP	WBIC	REACT	F-NP	FLAP
10kg	40	20	20	**5**	-	0.0111	**0.0108**	0.0110	-	0.5391	**0.3528**	0.3691	-	0.0957	**0.0793**	0.0806
7kg	27	5	3	**0**	0.0241	0.0108	0.0104	**0.0097**	0.9887	0.4691	0.3945	**0.3456**	0.2732	0.0874	0.0781	**0.0692**
4kg	0	0	0	0	0.0155	0.0094	0.0092	**0.0088**	0.6108	0.4241	0.3887	**0.3347**	0.1574	0.0757	0.0707	**0.0629**
Varied	38	14	9	**1**	0.0212	0.0110	0.0107	**0.0097**	1.1136	0.5262	0.4409	**0.3447**	0.2878	0.0969	0.0852	**0.0705**
Total Avg.	105	39	32	**6**	0.0181	0.0104	0.0102	**0.0098**	0.7399	0.4802	0.3985	**0.3481**	0.1967	0.0874	0.0779	**0.0708**
Avg. % Im.	-	62.86	69.52	**94.29**	-	42.54	43.65	**45.86**	-	35.10	46.14	**52.95**	-	55.57	60.40	**64.01**

IV. EXPERIMENTS

In this section, we describe our FLAP implementation and experimental setups, followed by the results obtained from both robotics simulations and real-world quadruped robot experiments conducted in unstructured field environments.

A. Experimental Settings

We experimentally validate FLAP through simulated and physical robot experiments using the Unitree Go1 quadruped robot. The quadruped locomotion controller $\Phi(\cdot)$ is implemented using the Whole Body Impulse Control (WBIC) method [1]. $\Psi(\cdot)$ is implemented as an MLP with the architecture shown in Fig. 2.

FLAP's training data consists of 2.88 million timesteps (96 minutes) collected in MuJoCo simulations [50], encompassing six payload masses (0, 1.0, 2.5, 5.0, 7.5, and 10 kg) and diverse locomotion behaviors (rotations, forward/backward, and lateral movements) on flat terrain. To simulate the effects of a dynamic payload, we applied random torques (resampled at 10 Hz) scaled by payload mass to the robot's torso, creating ground reaction force variations ($\Delta \mathbf{f}^r$) characteristic of the shifting mass distributions of dynamic payloads. During training data collection, we imposed velocity limits on $\mathbf{b}^{ex}$ for payloads $\geq$ 5.0 kg due to $\Phi(\cdot)$'s uncompensated payload capacity; these limits were removed during evaluation (Table II). Training and simulation used an 8-core i9 workstation (64GB RAM), while real-time deployment leveraged an Nvidia Jetson Orin Nano installed on the robot. Complete implementation, training, and evaluation details are available in the supplementary material.

We benchmark FLAP against three approaches: (1) non-adaptive $\Phi(\cdot)$ (**WBIC**) [1], (2) reactive compensation (**REACT**, Eq. (4)), and (3) a physics agnostic variant (**F-NP**) using an identical architecture to FLAP but without PINN loss terms $\mathcal{L}^p_{wb}$ in Eq. (8) and $\mathcal{L}^p_{\dot{\mathbf{q}}_{t+1}}$ in Eq. (7). All the approaches explored in this work generated locomotion commands at 0.5 kHz. Each of the adaptive methods share the same underlying locomotion controller $\Phi(\cdot)$ and BGF update law generating compensation values at 1 kHz. The values of the parameters α, κ, λ_0, and k_0 used by the BGF filter where set to 20.0, 1.2, 3.0, and 30.0 respectively and where held constant across all adaptive approaches and experimental settings. Performance is quantified through failure rates, Root Mean Squared Error (RMSE) of expected and actual torso pose/orientation and leg positions, expected torso velocity tracking accuracy, and percentage improvements over the WBIC baseline.

TABLE II

PAYLOAD $\mathbf{b}^{ex}$ LIMITS DURING TRAINING AND EVALUATION

Training		Evaluation	
Payload	$\mathbf{b}^{ex}$ Limits	Payload	$\mathbf{b}^{ex}$ Limits
$<$ 5kg	$\pm[1.0, 0.8, 2.0]$	$<$ 5kg	$\pm[1.0, 0.8, 2.0]$
$\geq$ 5kg	$\pm[0.5, 0.4, 1.0]$	$\geq$ 5kg	$\pm[1.0, 0.8, 2.0]$

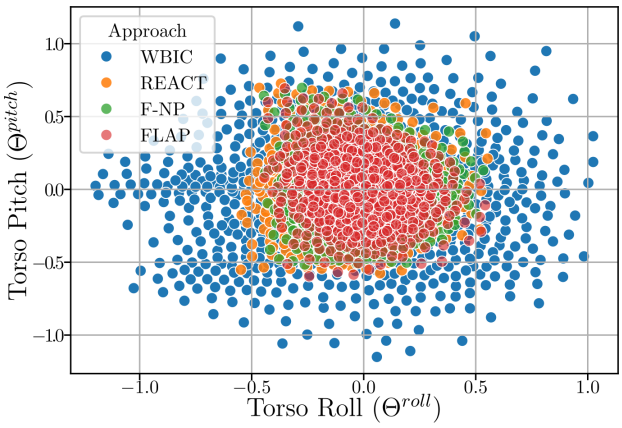


Fig. 3. Scatter plot of gravity projections through torso roll (Θ^{roll}) and pitch (Θ^{pitch}) capturing COM-shift experienced by each approach transporting a 4kg payload.

B. Results on Simulated Evaluation

We evaluated FLAP and baseline methods in MuJoCo simulations across four payload configurations (4kg, 7kg, 10kg, and 2-10kg variable mass) executing four distinct locomotion behaviors (forward/backward, lateral motion, turning, turning while moving) on flat terrain. Performance metrics were computed by averaging results from ten independent trials per condition, with final values aggregated by payload category for successful trials. Each successful trial comprised 23,493 timesteps (23.493 seconds) of continuous operation sampled at 1 kHz.

1) Tracking Performance: Table I presents the torso and leg joint pose tracking performance and failure rates. Overall, it was found that the forward looking adaptation variants outperformed the non-adaptive and reactive baselines across all conditions, with a key highlight being FLAP's 94.29% overall failure rate improvement over the WBIC baseline.

Moreover, FLAP outperforms the non-PINN F-NP method in terms of the pose tracking performance across all conditions other than for the 10 kg payload. In the later condition, the significant difference in failure rate between F-NP and FLAP, 50% and 12.5% respectively, suggests this minor performance disparity is a result of the stochasticity of the dynamic payload's random torso torques rather than controller performance.

Additionally, Fig. 3 depicts the gravity vector projected through the torso's orientation for each approach while transporting the 4 kg payload. The projected gravity vector captures how much the COM shifts due to the torso's roll and pitch, capturing the orientation stability of the torso throughout transport. As can be seen, FLAP results in projections that are more centered around 0 than the other approaches. These results collectively demonstrate FLAP's advantages for dynamic payload transportation over the baseline approaches.

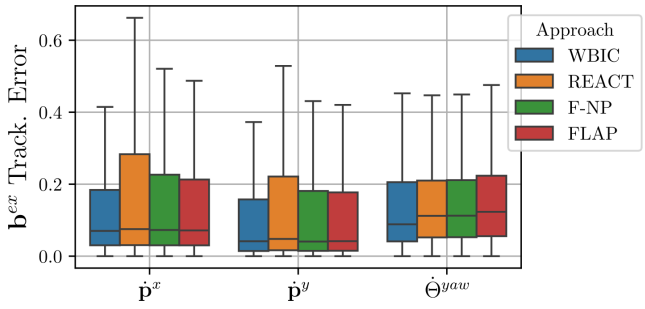


Fig. 4. Tracking error between expected $\mathbf{b}^{ex}$ and actual torso velocity behaviors ($\dot{\mathbf{p}}^x$, $\dot{\mathbf{p}}^y$, $\dot{\Theta}^{yaw}$).

Fig. 4 summarizes expected velocity command tracking across all conditions. FLAP achieves improved linear velocity tracking compared to adaptive baselines, but marginally worse yaw performance. Notably, all adaptive methods underperform the non-adaptive WBIC. This is because while compensating for payload disturbances improves torso stability/posture, it also induces slight velocity reductions during corrective phases. Proactive methods narrow the performance gap with WBIC in linear velocity tracking because their anticipatory adjustments require less severe corrections than reactive approaches. However, the enhanced torso stability during turns comes at the cost of reduced turning rates. These results demonstrate a minor trade-off between pose and velocity tracking performance, especially during turning maneuvers.

2) Forward Looking Prediction Performance: Fig. 5 compares the $\hat{\mathbf{q}}_{t+1}$ prediction accuracy of FLAP and F-NP under payload and $\mathbf{b}^{ex}$ conditions beyond the training distribution (where $\mathbf{b}^{ex}$ limits were relaxed per Table II). While F-NP's performance degrades with increasing payload and $\mathbf{b}^{ex}$-induced distribution shift, FLAP maintains stable prediction accuracy through its physics informed loss functions. This demonstrates that learning PINN-based physically consistent representations enables reliable generalization to novel system states, supporting the advantage of physics informed training.

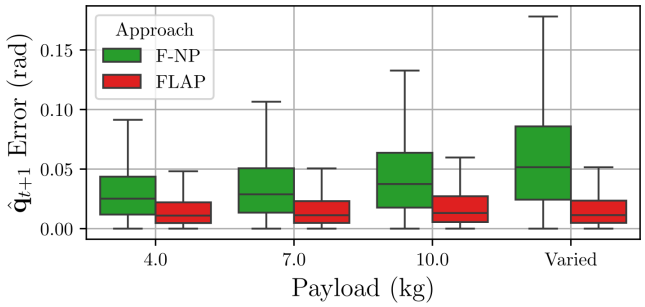
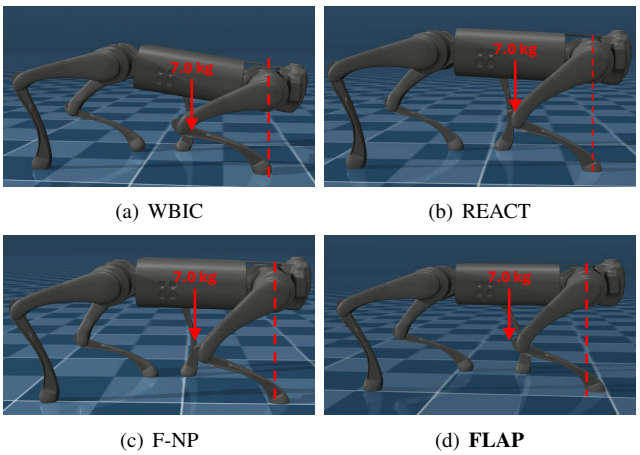


Fig. 5. FLAP and F-NP $\hat{\mathbf{q}}_{t+1}$ prediction errors vs. payload condition.

3) Qualitative Assessment of Simulated Results: Fig. 6 illustrates snapshots from successful trials of each method transporting a 7.0 kg dynamic payload at 1.0 m/s forward velocity. The non-adaptive WBIC baseline exhibits severe forward pitching, as the payload mass causes premature ground contact by the front right foot—placing it behind the centerline of the thigh rotor. This suboptimal foot position limits the robot's ability to generate sufficient counteracting forces/torques for torso stabilization. The reactive baseline shows reduced forward pitching, with improved foot placement partially ahead of the thigh rotor centerline. The F-NP approach further mitigates pitching, positioning most of the foot ahead of the centerline to better support the torso. In contrast, FLAP nearly eliminates forward pitching by placing the entire foot anterior to the thigh rotor centerline. This improvement stems from FLAP 's proactive compensation mechanism, which predicts joint configurations leading to early ground contact and preemptively increases adaptive gains to drive the foot into optimal contact position. The superior performance of FLAP over F-NP arises from its physics-informed training, which enables more accurate next-state predictions under novel dynamic conditions.



(a) WBIC (b) REACT

(c) F-NP (d) **FLAP**

Fig. 6. Each approach transporting a 7.0 kg virtual dynamic payload while moving forwards and 1.0 m/s. Each image contains a vertical red-dotted line that indicates the center of the front-right legs thigh rotor.

C. Real-World Liquid Payload Adaptation Demonstration

We validated FLAP 's robustness to real-world conditions through deployment on a physical robot carrying a 5 liter

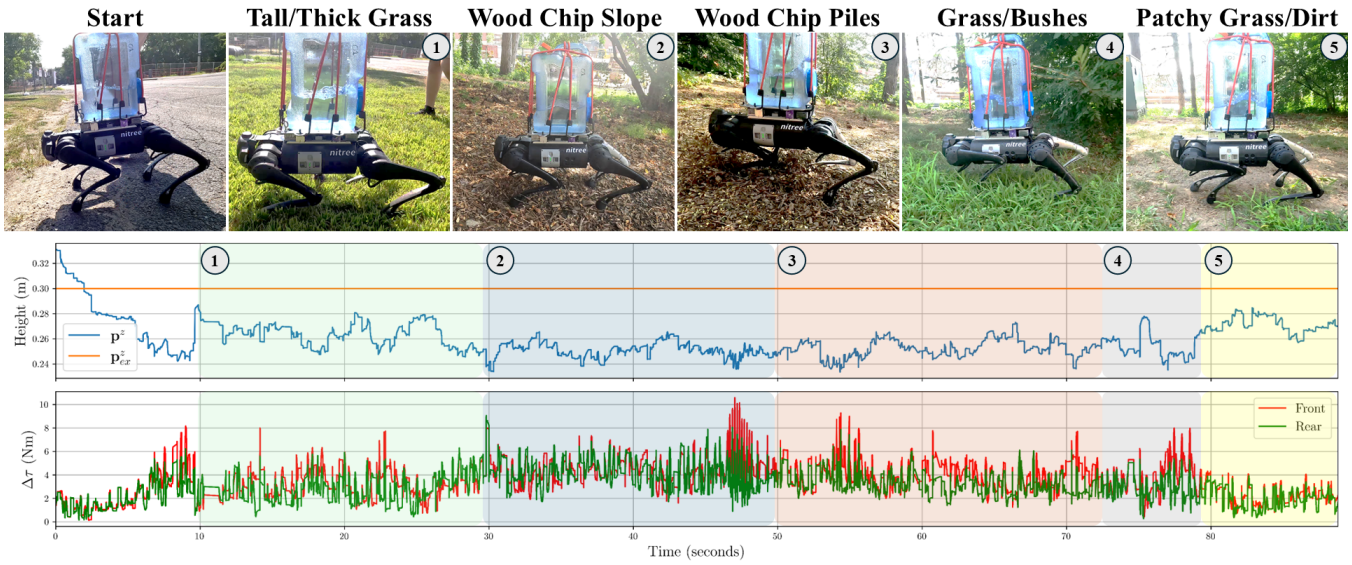


Fig. 7. Torso height tracking performance and the L2 norm of forward looking joint torque compensations (split between front and rear legs) of FLAP while transporting a 5 liter liquid water payload in real-world unstructured terrains.

liquid payload across a varied outdoor course featuring five distinct terrains not encountered during training. Details of the terrains encountered on the path and the robot's performance can be seen in Fig. 7. Transporting a liquid payload in the real-world requires not only counteracting the impact of the dynamic payload but also the ground reaction force variations introduced from unfamiliar terrains. During this evaluation, the robot began on flat pavement before progressing to an uneven lawn where its feet sank into thick grass. It then encountered a walking path with a 13.67° incline covered with loose wood chips. The incline disrupted expected foothold placements and caused the robot to pitch forwards. This necessitated substantial compensatory joint torques ($\Delta\tau$), particularly in the front legs. Subsequent sections included large wood chip piles that induced stumbling, tall grass with overhanging bushes that interfered with the payload, and finally patchy mixed terrain. Over the entire 65.45-meter course, completed in 88.79 seconds, FLAP successfully adapted to both the dynamic liquid payload and unpredictable terrain-induced disturbances. In contrast, the baseline controller $\Phi(\cdot)$ alone failed at the first terrain transition, underscoring the limitations of non-adaptive methods in complex real-world conditions. Complete videos of this and more outdoor experiments can be found on the project's website.

V. Conclusion

We presented FLAP, a physics-informed learning framework for dynamic payload adaptation in quadrupedal locomotion. Through novel PINN-based dynamics modeling and composite adaptive control, FLAP achieves superior performance compared to both reactive and proactive baselines. FLAP achieves a 94.29% lower failure rates than non-adaptive controller transporting dynamic payloads, with a 24.77% improvement over the next best approach. Additionally, experimental results demonstrate effective generalization to novel system states and previously unseen real-world terrains through learning physically consistent representations of the robot's dynamics. While revealing an inherent stability-velocity tradeoff during turning maneuvers, our results establish that embedding known dynamics into the learning process enables robust performance in real-world dynamic payload transportation scenarios. Future work will explore extending this framework to more complex terrain and payload configurations through incorporating PINN training when learning the underlying locomotion controller. Through including PINN loss functions in an RL training routine, we aim to reduce training data requirements while increasing the learned policies adaptability to novel conditions.

References

[1] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[3] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *Robotics: Science and Systems*, 2021.

[4] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.

[5] Y. Fan, Z. Pei, C. Wang, M. Li, Z. Tang, and Q. Liu, "A review of quadruped robots: Structure, control, and autonomous motion," *Advanced Intelligent Systems*, vol. 6, no. 6, p. 2300783, 2024.

[6] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *Field and Service Robotics*, vol. 16. Springer, 2021, pp. 247–260.

[7] S. Halder, K. Afsari, J. Serdakowski, S. DeVito, M. Ensafi, and W. Thabet, "Real-time and remote construction progress monitoring with a quadruped robot using augmented reality," *Buildings*, vol. 12, no. 11, p. 2027, 2022.

[8] K. Afsari, S. Halder, R. King, W. Thabet, J. Serdakowski, S. DeVito, M. Ensafi, and J. Lopez, "Identification of indicators for effectiveness evaluation of four-legged robots in automated construction progress monitoring," in *Construction Research Congress*, 2022.

[9] S. Halder, K. Afsari, E. Chiou, R. Patrick, and K. A. Hamed, "Construction inspection & monitoring with quadruped robots in future human-robot teaming: A preliminary study," *Journal of Building Engineering*, vol. 65, p. 105814, 2023.

[10] R. Siegwart, M. Hutter, P. Oettershagen, M. Burri, I. Gilitschenski, E. Galceran, and J. Nieto, "Legged and flying robots for disaster response," in *World Engineering Conference and Convention (WECC)*, 2015.

[11] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droeschel, and S. Behnke, "Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[12] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[13] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.

[14] G. Tournois, M. Focchi, A. Del Prete, R. Orsolino, D. G. Caldwell, and C. Semini, "Online payload identification for quadruped robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[15] M. Sombolestan, Y. Chen, and Q. Nguyen, "Adaptive force-based control for legged robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[16] M. V. Minniti, R. Grandia, F. Farshidian, and M. Hutter, "Adaptive clf-mpc with application to quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 565–572, 2021.

[17] M. Sombolestan and Q. Nguyen, "Adaptive-force-based control of dynamic legged locomotion over uneven terrain," *IEEE Transactions on Robotics*, vol. 40, pp. 2462–2477, 2024.

[18] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[19] Z. Zhuang, Z. Fu, J. Wang, C. G. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning*, 2023.

[20] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, and J. Tan, "Rapidly adaptable legged robots via evolutionary meta-learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[21] L. Smith, Y. Cao, and S. Levine, "Grow your limits: Continuous improvement with real-world rl for robotic locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[22] L. Amanzadeh, T. Chunawala, R. T. Fawcett, A. Leonessa, and K. A. Hamed, "Predictive control with indirect adaptive laws for payload transportation by quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10 359–10 366, 2024.

[23] C. Cao and N. Hovakimyan, "Design and analysis of a novel l1 adaptive control architecture with guaranteed transient performance," *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 586–591, 2008.

[24] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[25] Y. Song, S. bae Kim, and D. Scaramuzza, "Learning quadruped locomotion using differentiable simulation," in *Conference on Robot Learning*, 2025.

[26] Y. Chen and Q. Nguyen, "Learning agile locomotion and adaptive behaviors via rl-augmented mpc," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[27] X. Zeng, H. Zhang, L. Yue, Z. Song, L. Zhang, and Y.-H. Liu, "Adaptive model predictive control with data-driven error model for quadrupedal locomotion," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[28] S. Lyu, X. Lang, H. Zhao, H. Zhang, P. Ding, and D. Wang, "Rl2ac: Reinforcement learning-based rapid online adaptive control for legged robot robust locomotion," *Robotics: Science and Systems (RSS)*, 2024.

[29] O. Youngquist and H. Zhang, "Rina: Rapid introspective neural adaptation for out-of-distribution payload configurations on quadruped robots," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 15 972–15 979.

[30] V. K. Kurva and S. Kolathaya, "Mule: Multi-terrain and unknown load adaptation for effective quadrupedal locomotion," *arXiv preprint arXiv:2505.00488*, 2025.

[31] Y. Zhang, B. Nie, Z. Cao, Y. Fu, and Y. Gao, "Disturbance-aware adaptive compensation in hybrid force-position locomotion policy for legged robots," *arXiv preprint arXiv:2506.00472*, 2025.

[32] 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," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[33] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020.

[34] C. Zhao, F. Zhang, W. Lou, X. Wang, and J. Yang, "A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics," *Physics of Fluids*, vol. 36, no. 10, 2024.

[35] D. Jalili, S. Jang, M. Jadidi, G. Giustini, A. Keshmiri, and Y. Mahmoudi, "Physics-informed neural networks for heat transfer prediction in two-phase flows," *International Journal of Heat and Mass Transfer*, vol. 221, p. 125089, 2024.

[36] H. Hu, L. Qi, and X. Chao, "Physics-informed neural networks (pinn) for computational solid mechanics: Numerical frameworks and applications," *Thin-Walled Structures*, vol. 205, p. 112495, 2024.

[37] L. Wang, G. Liu, G. Wang, and K. Zhang, "M-pinn: A mesh-based physics-informed neural network for linear elastic problems in solid mechanics," *International journal for numerical methods in engineering*, vol. 125, no. 9, p. e7444, 2024.

[38] J. Liu, P. Borja, and C. Della Santina, "Physics-informed neural networks to model and control robots: A theoretical and experimental investigation," *Advanced Intelligent Systems*, vol. 6, no. 5, p. 2300385, 2024.

[39] W. Deng, F. Ardiani, K. T. Nguyen, M. Benoussaad, and K. Medjaher, "Physics informed machine learning model for inverse dynamics in robotic manipulators," *Applied Soft Computing*, vol. 163, p. 111877, 2024.

[40] S. Sanyal and K. Roy, "Ramp-net: A robust adaptive mpc for quadrotors via physics-informed neural network," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[41] W. Gu, S. Primatesta, and A. Rizzo, "Physics-informed neural network for quadrotor dynamical modeling," *Robotics and Autonomous Systems*, vol. 171, p. 104569, 2024.

[42] M. Lahariya, C. Innes, C. Develder, and S. Ramamoorthy, "Learning physics-informed simulation models for soft robotic manipulation: A case study with dielectric elastomer actuators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[43] T.-L. Habich, A. Mohammad, S. F. Ehlers, M. Bensch, T. Seel, and M. Schappler, "Generalizable and fast surrogates: Model predictive control of articulated soft robots using physics-informed neural networks," *arXiv preprint arXiv:2502.01916*, 2025.

[44] X. Yang, Y. Du, L. Li, Z. Zhou, and X. Zhang, "Physics-informed neural network for model prediction and dynamics parameter identification of collaborative robot joints," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8462–8469, 2023.

[45] I. Sorrentino, G. Romualdi, L. Moretti, S. Traversaro, and D. Pucci, "Physics-informed neural networks with unscented kalman filter for sensorless joint torque estimation in humanoid robots," *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5705–5712, 2025.

[46] R. Ni and A. H. Qureshi, "Physics-informed neural motion planning on constraint manifolds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[47] Y. Liu, R. Ni, and A. H. Qureshi, "Physics-informed neural mapping and motion planning in unknown environments," *IEEE Transactions on Robotics*, vol. 41, pp. 2200–2212, 2025.

[48] J.-J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[49] Y. Pan and H. Yu, "Composite learning robot control with guaranteed parameter convergence," *Automatica*, vol. 89, pp. 398–406, 2018.

[50] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.