2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Toward Control of Wheeled Humanoid Robots with Unknown Payloads: Equilibrium Point Estimation via Real-to-Sim Adaptation

Donghoon Baek[1], Youngwoo Sim[1], Amartya Purushottam[2], Saurabh Gupta[2,3], and Joao Ramos[1,2]

***Abstract*— Model-based controllers using a linearized model around the system's equilibrium point is a common approach in the control of a wheeled humanoid due to their less computational load and ease of stability analysis. However, controlling a wheeled humanoid robot while it lifts an unknown object presents significant challenges, primarily due to the lack of knowledge in object dynamics. This paper presents a framework designed for predicting the new equilibrium point explicitly to control a wheeled-legged robot with unknown dynamics. We estimated the total mass and center of mass of the system from its response to initially unknown dynamics, then calculated the new equilibrium point accordingly. To avoid using additional sensors (e.g., force torque sensor) and reduce the effort of obtaining expensive real data, a data-driven approach is utilized with a novel real-to-sim adaptation. A more accurate nonlinear dynamics model, offering a closer representation of real-world physics, is injected into a rigid-body simulation for real-to-sim adaptation. The nonlinear dynamics model parameters were optimized using Particle Swarm Optimization. The efficacy of this framework was validated on a physical wheeled inverted pendulum, a simplified model of a wheeled-legged robot. The experimental results indicate that employing a more precise analytical model with optimized parameters significantly reduces the gap between simulation and reality, thus improving the efficiency of a model-based controller in controlling a wheeled robot with unknown dynamics.**

I. Introduction

Wheel-legged robots have emerged as a potential platform to facilitate operations in a wide range of industries, including agriculture, mining, military, and search and rescue. Combining the mobility of wheels and the mobility of legs, the capability to navigate difficult environments could be greatly improved. [1], [2], [3], [4], [5].

Controlling a wheeled-legged robot poses significant challenges due to its inherent instability and complex nonlinear dynamics. Traditional methods often simplify the system by linearizing the system dynamics around its equilibrium point, enabling the use of diverse control techniques like the Linear Quadratic Regulator (LQR) or Model Predictive Control (MPC) [1], [6]. The underlying assumption is that the system can be treated as a linear system within the equilibrium point. For example, within the equilibrium point, the inverted pendulum remains upright and stationary, meaning that the net force acting on the pendulum is zero and it does not experience any acceleration. However, when the robot engages in tasks like manipulating objects or pushing heavy loads, these changes can drastically alter the equilibrium point due to shifts in the center of mass (CoM) or contact forces, making linear approximations inadequate for effective control [7]. Recently, approaches such as deep reinforcement learning (RL) and domain randomization have emerged, showing a promising result in locomotion tasks [8], [9], [10]. Yet, they suffer from interpretability issues and cannot assure stability.

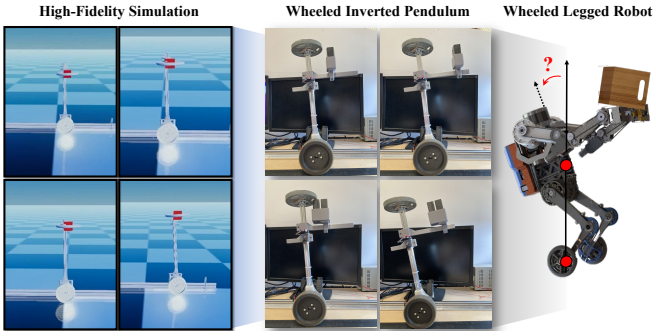


Fig. 1: **Conceptual overview of the proposed method.** The data-driven regression model learns to identify the new equilibrium point for a wheeled inverted pendulum from a high-fidelity simulation closely mirroring the real world. A physical version of this pendulum tests the framework's viability, suggesting its applicability in controlling a wheeled humanoid robot lifting an unknown object.

In this work, our focus is on addressing the challenge of a wheeled humanoid robot transporting an unseen object during locomotion. The framework capable of rapidly and explicitly estimating the new equilibrium point of a wheeled-legged robot is proposed. The main idea is to directly identify the equilibrium point by capturing the momentary change in a history of initial proprioceptive states when the robot falls (e.g., the heavier and farther away the object, the faster the robot will fall). Our approach does not involve solving the Newton-Euler equations, which are typically employed in system identification processes. As a result, we avoid the need for using force/torque sensors and cameras to estimate the new center of mass and total mass when an object is included. Moreover, our framework can simply be incorporated with a classical model-based controller without re-formulating their equation.

A prominent challenge in adopting data-driven approaches is the issue of data scarcity. To counter this, we built a high-fidelity simulation via real-to-sim adaptation and trained a data-driven model in the simulation. The adaptation method includes a more accurate dynamics model that accounts for friction, damping, and actuator dynamics, elements that are often oversimplified in typical rigid-body dynamics simulations.

This work is supported by the National Science Foundation via grant IIS-2024775.

The authors are with the [1] Department of Mechanical Science and Engineering and the [2] Department of Electrical and Computer Engineering and the [3] Department of Computer Science at the University of Illinois at Urbana-Champaign, USA.dbaek4@illinois.edu

II. Related Work

Wheeled-Legged Robot Control: The most popular approach to control wheeled-legged robots is model-based control, which leverages a mathematical dynamics model. Some strategies use simplified models, like the Wheeled Inverted Pendulum (WIP), and linearize them around the system's equilibrium point [3], [2], [5]. To achieve better tracking and versatility, more complex models have been applied [1], [11], [12]. However, the success of these methods depends greatly on the precision of the modeled dynamics, a precision that is difficult to attain and may not adjust well to changes in the system or environment. As an alternative, adaptive controllers have been presented to enable the robots to adapt to new environments [13], [14]. Moreover, learning-based methods, combining deep neural networks with reinforcement learning (RL), have been suggested as an effective approach to address the nonlinear locomotion challenges of legged robots [9], [8]. With the benefits of simulation for safe and efficient training, many RL studies employ sim-to-real techniques, resulting in successfully transferring the RL policy to the real world [15], [16], [17]. Despite their innovative success, they often require a lot of manual tuning or have many difficulties in interpreting the behavior and ensuring the safety of the system.

Sim-to-Real Transfer: Despite learning-based approaches making controller design in simulations simpler and less reliant on specialized domain knowledge, sim-to-real transfer still necessitates extensive manual adjustments. For example, the process of choosing dynamic parameters to randomize for training a robust RL policy across diverse dynamics remains complex and not easily automated [16], [17], [18]. An alternative way to reduce a *reality gap* is system identification which entails adjusting model parameters to align the simulation's observations with those from actual hardware [19], [20], [21], [22], [23]. While this approach has helped generate robust policies, the focus lies solely on adapting the physics engine's parameters or distribution which are often inaccurate due to their simplification. Many rigid-body simulators prioritizing fast computation often employ simplified dynamics to lessen computational demands, typically overlooking factors like static and viscous friction at joints, damping, and actuator dynamics. While some studies have explored data-driven approaches to model actuator dynamics [10], [15], these are not universally applicable to passive joints or varied controller types, such as position or torque modes. Instead of merely adjusting existing simulator parameters, our approach integrates a more detailed dynamics model with its optimized parameters into the simulation, aiming to minimize the *reality gap*.

III. BACKGROUND

The qualitative behavior of the nonlinear system in the vicinity of a hyperbolic equilibrium point is determined by the characteristics of the corresponding linear system [24]. Given that $x_0 = 0$ is an equilibrium point of the system, we have $f(0) = 0$. By applying Taylor's Theorem, we can express the function $f(x)$in the expanded form as follows:

$$\dot{x} = f(x) = Df(0)x + \frac{1}{2}D^2f(0)(x,x) + \dots \qquad (1)$$

where $Df(0)x$ denotes the Jacobian matrix of f evaluated at 0, linearly acting upon x, and $D^2f(0)(x,x)$ indicates the higher-order terms. The same theory can be applied to a wheeled-humanoid robot (e.g., SATYRR [2]) to linearize its nonlinear dynamics. The wheeled inverted pendulum is commonly taken to express the motion of a wheeled humanoid and its equation of motion is as follows:

$$\begin{aligned}\left(m_b+m_w+\frac{I_w}{r^2}\right)\ddot{x}_w+m_bLs(\theta)\dot{\theta}^2-m_bLc(\theta)\ddot{\theta}=u\\ (m_bL^2+I_b)\ddot{\theta}-m_bLc(\theta)\ddot{x}_w-m_bgLs(\theta)=0\end{aligned} \qquad (2)$$

where $m_b, m_w, L, \theta, I_w, I_b, r,$ and u denote the body and wheel masses, the distance between the CoM to the center of the wheel, pitch angle, wheel and body inertia, wheel radius, and control force, respectively. Linearizing the system's dynamics at the equilibrium (CoM above the wheel) yields a linear state-space model:

$$\Delta\dot{\mathbf{q}} = \mathbf{A}\Delta\mathbf{q} + \mathbf{B}\Delta u, \qquad (3)$$

where $\mathbf{q} = [x_w, \theta, \dot{x}_w, \dot{\theta}]^\top$ defines the state vector, and $\mathbf{A} \in \mathbb{R}^{4\times4}$, $\mathbf{B} \in \mathbb{R}^{4\times1}$ are state-space matrices. Deviations from equilibrium state $\mathbf{q_0}$ and nominal control effort u_0 are $\Delta\mathbf{q} = \mathbf{q} - \mathbf{q_0}$ and $\Delta u = u - u_0$, respectively. Changes in total mass and equilibrium point θ_0, such as when lifting an object, affect these dynamics, degrading model-based controller performance. An LQR controller π_H designed with the above dynamics serves as our baseline for evaluating the proposed framework.

IV. Method

A. Real-to-Sim Adaptation via a High-Fidelity Simulation

Unlike previous studies that apply the domain transfer to a trained model to bridge the domain gap $\Delta\mathbf{x}$ during or after training, our approach aims to reduce the *reality gap* at the beginning of the entire process. This can be achieved by incorporating a more precise dynamics model that accounts for nonlinear friction, damping, and motor dead-zone effects into a rigid-body simulation. Especially, in this work, we are interested in minimizing the gap caused by the parametric modeling error $\Delta\mathbf{x_p}$.

$$\Delta\mathbf{x}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t) = \underbrace{\Delta\mathbf{x_p}(\boldsymbol{\zeta}, t)}_{\text{Parametric Error}} + \underbrace{\Delta\mathbf{x_{np}}(\mathbf{x}, t)}_{\text{Nonparametric Error}} \qquad (4)$$

Here, $\Delta\mathbf{x}(t)$ denotes the *reality gap*, with $\Delta\mathbf{x_p}$ as the parametric error dependent on model parameters $\boldsymbol{\zeta}$ and time t, and $\Delta\mathbf{x_{np}}$ representing nonparametric error related to state $\mathbf{x}$ and time t. Particle Swarm Optimization (PSO)[25] is used for parameter optimization. A schematic of the framework is depicted in Fig. 2.

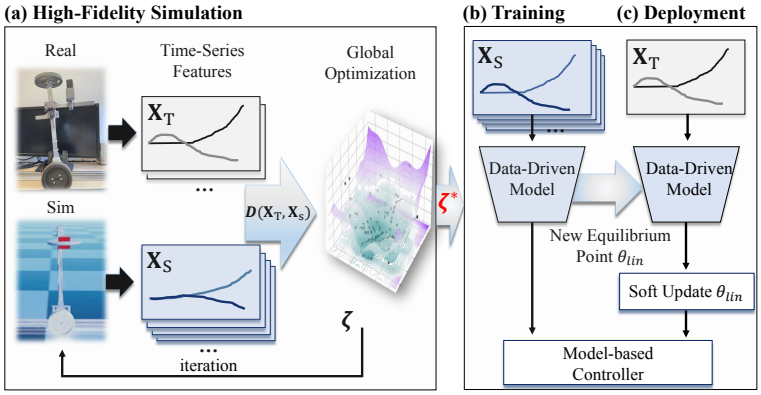


Fig. 2: **Real-to-Sim Adaptation via a High Fidelity Simulation.** (a) High-fidelity simulation is achieved by minimizing parametric modeling error, $\Delta \mathbf{x_p}$, via updates to $\boldsymbol{\zeta}$. The state trajectories of the system in two different domains are obtained from a physical system and a simulation, respectively, offline. The global optimization algorithm (e.g., PSO) is utilized to identify the parameters $\boldsymbol{\zeta}$ to deduce the *reality gap*. (b) A data-driven model (e.g., LSTM) is trained to predict the new equilibrium point via supervised learning with the data obtained from a high-fidelity simulation. (c) The trained data-driven model estimates the new equilibrium point online in deployment.

Data Construction. To build a high-fidelity simulator toward achieving a real-to-sim adaptation, we constructed the M number of source dataset $\mathcal{D}_{\mathcal{S}} = ({\mathbf{X_S}}^i, y_S^i)_{i=1}^M$, and a small m number of target dataset $\mathcal{D}_{\mathcal{T}} = ({\mathbf{X_T}}^i, y_T^i)_{i=1}^m$ obtained from a simulator and a real-world, respectively ($m = 40, M = 1200$). Both domains acquire samples drawn from the source and target distribution, $\mathcal{D}^s \sim p_s(\mathbf{X_S}, y_S)$ and $\mathcal{D}^t \sim p_t(\mathbf{X_T}, y_T)$ which typically differ ($\mathcal{D}^s \in \mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}^t \in \mathcal{D}_{\mathcal{T}}$). In our application, the output y indicates the linearized pitch angle θ deciding the equilibrium point. Inspired by the previous research [17], the input data $\mathbf{X}$ consists of the state vector $\mathbf{X} = [x_{t:T}, \theta_{t:T}]^\top$ represents a T time-series trajectory of linear position and pitch angle. Since the goal is to estimate the new equilibrium point of a wheeled-legged robot in a situation where the total payload and CoM position change, we randomized the total mass, inertia, and CoM position of the robot while collecting the dataset $\mathcal{D}$. We utilized a RaiSim [26] simulation to construct the simulation dataset $\mathcal{D}_S$. Note that the new angle θ_{lin} is easily computed in a simulation with the known position of CoM. In the real world, we manually measured the θ_{lin} at which the robot can stabilize itself without controller intervention (see Fig. 5). To get more accurate and less noisy data in the real world, we applied the Extended Kalman Filter to a physical system and utilized the same control gain in both domains.

More Accurate Nonlinear Dynamic Function. The nonlinear dynamics model is designed by combining the core elements typically factored in during a sim-to-real transfer [15], [16], [17]. Since the success of most machine learning models (e.g., Gaussian Processes, Deep Neural Networks) highly depends on the quantity and quality of the dataset, we adopted to use of an analytical model with its optimized parameters considering the sample efficiency to reduce the *reality gap*. Although some rigid-body simulators have a built-in function allowing users to tune the parameter of friction and damping, their function is usually simplified to reduce the complexity and sometimes not clear in deciding the numerical value. To use a more accurate and intuitive dynamic model in a simulation, we designed the nonlinear dynamics model that takes account of the effect of the static friction, damping, latency, and actuator dynamics in each joint. Here, a more accurate is denoted by $g_{\boldsymbol{\zeta}}$ which satisfies $\mathbf{x_{t+1}} = f(\mathbf{x_t}, \mathbf{u_t}, g_{\boldsymbol{\zeta}}(\mathbf{x_t}))$, where $\boldsymbol{\zeta} = (F_{ss}, F_c, vs, \omega, \epsilon, \alpha)$ is physics parameters of the model $g_{\boldsymbol{\zeta}}$. The nonlinear dynamics model $g_{\boldsymbol{\zeta}}$ is designed as follows:

$$
\begin{aligned}
g_{\boldsymbol{\zeta}}(t) &= -\alpha F_{ss}(t) + (\alpha - 1) F_{ss}(t-1) \\
F_{ss} &= \begin{cases} 0 & \dot{q} < \epsilon \\ F_c \text{sign}(v) + (F_s - F_c) e^r \text{sign}(v) + \sigma v & \dot{q} >= \epsilon \end{cases} \\
e^r &= -(v/vs)^2
\end{aligned}
\tag{5}
$$

where α represents the damping ratio, which delays signals, and F_{ss} the static and dynamic friction model, incorporating a dead zone effect denoted by ϵ. The friction model, F_{ss}, integrates Coulomb friction F_c, the differential static friction, and viscous damping σ, where the damping force correlates with velocity (see Fig. 3). The Stribeck velocity, vs, marks the shift from static to dynamic friction. Notably, the friction force, F_{ss}, is significant in steady states with minimal movement v,

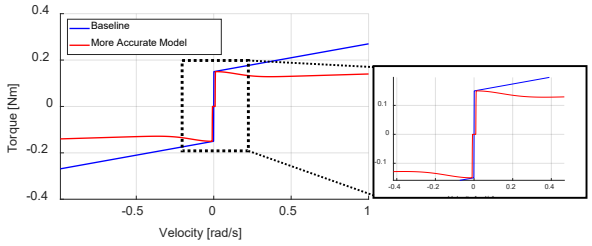


Fig. 3: **Visualizing the Nonlinear Dynamic Function $g_{\boldsymbol{\zeta}}$.** The selected parameter $\boldsymbol{\zeta}$ is [0.15, 0.12, 0.2, 0.02, 0.01, 0.7]. The model exhibits non-responsiveness near zero speed due to the dead zone effect, with a behavior that is more nonlinear compared to the standard viscous and Coulomb friction model (Baseline). Additionally, a slight phase shift is observed, indicative of a delay.

Algorithm 1 Real-to-Sim Adaptation Procedure

Input: $\mathcal{D}_\mathcal{T} = \{D_T^1, D_T^2, \ldots, D_T^m\}, m = 8$
Initialize: $\boldsymbol{\zeta} \in \mathbb{R}^k$, $\mathcal{L} \in \mathbb{R}, k = 12$ ▷ Initialize parameter of $g_{\boldsymbol{\zeta}}$ and Loss function
while until ($\mathcal{L}$ converges) **do**
  $(\mathbf{X_T}^i, y_T^i)_{i=1}^m \leftarrow \mathcal{D}_\mathcal{T}$ ▷ From a real-world
  for $i = 0, ...m$ **do** ▷ Run Raisim Simulator
    $(\mathbf{X}_S^i, y_S^i)_{i=1}^m \leftarrow \pi_H(\boldsymbol{\zeta})$ ▷ From a baseline LQR
  end for
  Calculate the Loss function $\mathcal{L}$ ▷ Eq. (7)
  Update parameter $\boldsymbol{\zeta}$ via *PSO algorithm*
end while
Return: $\boldsymbol{\zeta}^*$

leading to instability. To mitigate this, we introduce a dead-zone ϵ in Eq. (5). In our application, the model $g_{\boldsymbol{\zeta}}$ is specifically applied to translation and actuator joints separately to address high friction and latency issues. Moreover, we applied the artificial high-frequency noise to acquire more realistic observed state $\mathbf{X_S}$. The calculation for this high-frequency noise denoted as w, is presented subsequently.

$$w = Acos(2\pi ft) \times B\mathcal{N}(0, 1) \tag{6}$$

where A, B, and variables $f, t, \mathcal{N}(0, 1)$ representing frequency, time, and a normally distributed random number, respectively (setting $A = 0.01, B = 0.5, f = 0.0002$), noise w is added to the original observation. We also chose an inertia value more carefully bu calculating the moment of inertia (e.g., $I = ML^2$) rather than using an arbitrarily selected value within a specific range.

Parameter Optimization For Nonlinear Function via Particle Swarm Optimization. If the accumulation of model match over trajectories $\mathbf{X_S^t}$ and $\mathbf{X_T^t}$ is the same, we can say that there is almost no *reality gap* in two domains. Therefore, the objective is to find the most appropriate parameters $\boldsymbol{\zeta}$ resulting in matching two trajectories as close as possible. We leverage mean square error (MSE) as a distance metric $d(\cdot|\cdot)$ across m samples, each spanning T units, to effectively bridge the *reality gap* [20].

$$\arg\min_{\zeta} \frac{1}{m} \sum_{i=1}^{m} \sum_{t=1}^{T} d(\mathbf{X_S^t}, \mathbf{X_T^t}). \tag{7}$$

The optimal parameter $\boldsymbol{\zeta}^*$ can be chosen by solving the Eq. 7 and we utilized the PSO algorithm due to its global search ability and fast convergence speed. Real-to-Sim adaptation procedure is described in Algorithm 1.

The PSO algorithm, implemented via the SciPy package [27], updates particle velocities and positions as per:

$$\begin{aligned} V_{id}^{t+1} &= wV_{id}^t + c_1 r_1 (P_{b,id}^t - X_{id}^t) + c_2 r_2 (G_{b,id}^t - X_{id}^t), \\ X_{id}^{t+1} &= X_{id}^t + V_{id}^{t+1} \end{aligned} \tag{8}$$

Here, V_{id}^{t+1} and X_{id}^{t+1} denote the ith particle's velocity and position in dimension d at time $t + 1$. Parameters include inertia weight w, and cognitive and social factors c_1 and c_2, influencing the particle's momentum and the pull towards its best and the swarm's best positions, respectively, with $P_{b,id}^t$ and $G_{b,id}^t$ representing these best positions in dimension d at time t ($c1 = 0.5, c2 = 0.2, w = 0.9$).

In the optimization of parameter $\boldsymbol{\zeta}$, the mean $\mathbf{X_T^i} \in \mathbb{R}^{N \times L}$ of real data per class is utilized, where $N = 2$ (features) and $L = 80$ (window size). The parameter range for $\boldsymbol{\zeta}$ in global optimization is manually tuned. For optimizing the built-in simulation parameters, seven parameters including joint friction and damping, crucial for *domain randomization*, are selected.

B. Estimating New Equilibrium Point via Data-Driven Model

Training a Data-Driven Model. To capture the new equilibrium point for a wheeled-legged robot with unknown dynamics, we focus on the initial behavioral characteristics of a robot under different dynamics influences (e.g., changes in payload weight and center of mass (CoM) position affecting speed during forward falls). This behavior is encapsulated as preconception history, employing time-series features $\boldsymbol{X} = [x_{t:N}, \theta_{t:N}]^\top$ as inputs to a data-driven estimator.

Distinct datasets were constructed from varied simulation configurations, each mirroring the specifics of corresponding baseline real-to-sim methods. A total of 1,500 data points constituted each dataset, divided into training, validation, and testing sets with an 8:1:1 split. The optimum model for each method, validated through simulation datasets, was chosen for benchmarking, as shown in Table I. The sequence length was established at 80, equivalent to approximately 1.2 seconds of data. The data-driven estimator, exemplified by an LSTM, underwent training exclusively on simulation-derived datasets. Specifically, the LSTM was trained over 500 epochs with a batch size of 256 and a learning rate of 1×10^{-3}, alongside a weight decay of 1×10^{-5}. The LSTM featured a hidden size of 1024 across two layers.

Deployment in a Physical System. The pre-trained data-driven model is utilized to predict the new equilibrium point in a physical system without using further manual tuning. The WIP system, when in motion, tends toward the direction where additional mass is added, as illustrated in Fig 5. The estimator quickly identifies the new equilibrium point within a short time frame (less than 1.2 sec). To facilitate a smooth transition in the desired pitch angle reference, a soft-update method is employed.

$$\begin{aligned} \theta^{des} &= max(\theta_{lin}, -\beta t) \\ \theta_t^{sm} &= \alpha \theta_t^{des} + (1 - \alpha)\theta_{t-1}^{sm} \end{aligned} \tag{9}$$

where the first equation encourages that the reference angle updates gradually until the new equilibrium point θ_{lin}. The role of the second equation is to update the reference angle smoothly such as polynomial trajectory. Hyperparameters α and β are manually tuned. ($\alpha = 0.05$ and $\beta = 0.1$)

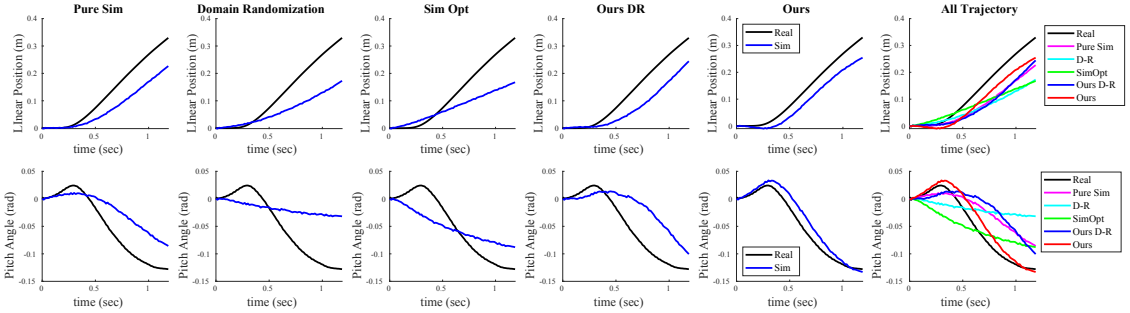


Fig. 4: **Sample Result of Data Trajectory Comparison Between a Simulation and the Real World.** The last graph presents a comparison of data trajectories between simulation and reality for all methods. Our approach effectively narrows the *reality gap* and aligns the trajectory shape with the actual one. The residual error is attributed from coupled-dynamics, non-parametric nonlinear dynamics, and class-specific data variance. All data samples are collected from the WIP system using LQR control.

V. Experiment

Three separate experiments are conducted: 1. Verification of real-to-sim adaptation in estimating the new equilibrium point in both a simulation and the real world. 2. Evaluation of the tracking performance of an LQR applied the newly equilibrium point. 3. An ablation study aimed at investigating the influence of the chosen time-series regression model and optimization algorithm.

A. Simulation Setting

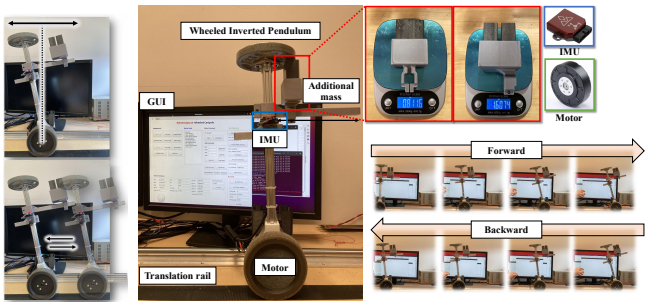
A RaiSim [26] simulation is used to collect the simulation dataset $\mathcal{D}_{\mathcal{S}}$. Unified Robot Description Format (URDF) [28] file is employed to simulate a customized wheeled inverted pendulum. The control loop runs at a control frequency of approximately 600-700Hz considering the hardware control frequency.

B. Target Prototype Hardware Testbed: Wheeled-Inverted Pendulum

A customized wheeled-inverted pendulum (WIP) was developed to verify the feasibility of the proposed framework as depicted in Fig. 5. The WIP is often used as a template model to control a wheeled-legged robot [3]. Note that while the degree of freedom of WIP is less complex, this still maintains intricate interactions between the robot and its surroundings, including contact dynamics distinct from those of a traditional inverted pendulum. Extra payloads can be affixed to alter its total payload and the position of the CoM. The same actuator equipped in the MIT Mini Cheetah [29] is employed and an inertial measurement unit (VN-100, VectorNav, USA) is mounted to the pole link. The motor communicates over the CAN bus with the desktop PC (Ubuntu20.04) and all software is connected via Robot Operating System (ROS).

C. Experimental Plan

1) Real-to-Sim Adaptation: The first experiment was conducted to assess the estimation performance of the data-driven estimator trained in a high-fidelity simulation that is achieved by our proposed framework. The objectives of the experiment are two-fold: (1) to analyze the benefits of a highly precise model in the context of real-to-sim adaptation,



(a) (b)

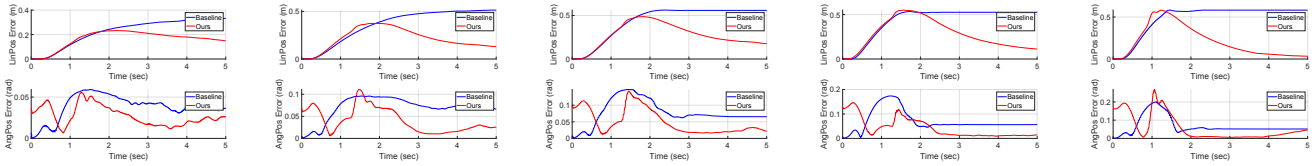
Fig. 5: **A Physical Wheeled Inverted Pendulum Testbed.** (a) Balancing (top) and tracking a sinusoidal reference task (bottom) were conducted in non-ideal and unknown dynamic situation. (b) A customized wheeled Inverted pendulum is made up of a motor, wheel, IMU, translation rail, and weight. Two additional weights (0.8kg, 1.6kg) can be attached and detached to a pole link to adjust the total mass and the position of the CoM.

and (2) to investigate how our approach stands against traditional domain randomization. We evaluate the impact of the high-fidelity simulation by contrasting it with four separate baselines: (a) **Pure Sim**: simulation using default physics parameters (b) **Ours WO Opt**: a more accurate dynamic model (Eq. 5) with random parameters applied within a simulation (c) **Sim Param Opt**: built-in simulation parameters adapted through a global optimization algorithm (d) **Sim Param D-R**: built-in simulation parameters are randomized (*domain randomization*). For the validation, the dataset we described in the section IV-A is utlized.

2) Balancing and Tracking Tasks: The second experiment was conducted to explore the impact of integrating the new equilibrium point into a model-based controller to enhance its tracking performance. The trained data-driven model effectively identified the new equilibrium point online at operation onset by observing the system's behavior under unknown dynamics. This equilibrium point was integrated into an LQR controller, enhancing its ability to track the reference signal and recover the original position of the system. Controller tracking performance was assessed with and without the new equilibrium point in two scenarios: balancing (Task 1) and periodic signal tracking (Task 2), where 'Ours' utilized the new equilibrium point as a target

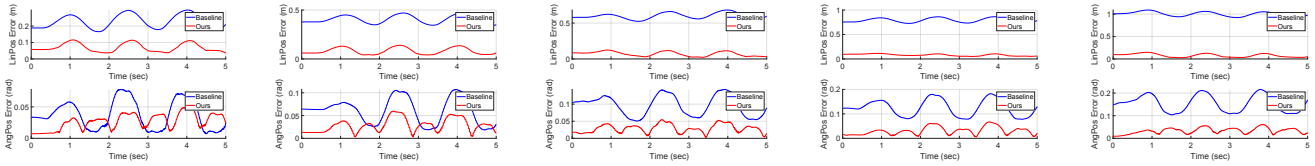
TABLE I: **Comparison Results of Estimating the Equilibrium Point.** Mean Square Error (MSE) from 146 test data in simulation (Sim) and 40 real-world (Real) test data, not involved in training, serves as the evaluation criteria. The discrepancy in MSE between a simulation (Sim) and real world (Real) assesses the model's capability to adapt simulation-trained data to real-world scenarios.

		Pure Sim		Ours WO Opt		Sim Param Opt		Sim Param D-R		**Ours**	
		Sim	Real	Sim	Real	Sim	Real	Sim	Real	Sim	Real
MSE (*rad*)	mean	0.012	0.033	0.011	0.030	0.089	0.096	**0.008**	0.111	0.024	**0.023**
	std	0.010	0.003	0.007	0.005	0.063	0.025	0.008	0.010	0.015	0.005
Difference		0.021		0.019		0.007		0.103		**0.001**	



(a) $\theta_{lin} = \ 0.03rad$ (b) $\theta_{lin} = \ 0.06rad$ (c) $\theta_{lin} = \ 0.09rad$ (d) $\theta_{lin} = \ 0.12rad$ (e) $\theta_{lin} = \ 0.16rad$

Fig. 6: **Results for Balancing Tasks in Several Scenarios with Different, Unknown Dynamics.** Graphs (a) to (e) display the absolute error between targeted and actual trajectories for both linear (x_w) and angular (θ) positions (mean of five trails). The baseline assumes zero for both desired positions, while our method employs the new equilibrium point as the targeted angular position. Each graph corresponds to a unique equilibrium point, reflecting varied payloads and the center of mass. The baseline LQR tends to fail, struggling with equilibrium points significantly deviated from the original (When the system reached and crashed the end of rail, the system halt at a certain point - this is why the baseline graph appears to stop at a certain point). We observed the system continuously fell until reaching its motion limit. Conversely, our method quickly identified the new equilibrium point (within approximately 1 second) and effectively recovered its position in all different cases.



(a) $\theta_{lin} = \ 0.03rad$ (b) $\theta_{lin} = \ 0.06rad$ (c) $\theta_{lin} = \ 0.09rad$ (d) $\theta_{lin} = \ 0.12rad$ (e) $\theta_{lin} = \ 0.16rad$

Fig. 7: **Results for Tracking Tasks in Several Scenarios with Different, Unknown Dynamics.** Graphs (a) to (e) show the same criteria we mentioned in Fig. 6 for the tracking task. The periodic (sinusoidal) signal is utilized as a desired reference. Incorporating a new equilibrium point as the desired angular position, our method demonstrated superior tracking accuracy for both linear and angular positions across all cases.

angular position, and 'Baseline' did not. We adjusted the total mass and CoM position of the WIP by adding weights to the pole link, simulating a wheeled-legged robot transporting an unknown object. Two task experiments were conducted across five different cases, with each case undergoing five trials.

VI. Result and Discussion

A. Experiment 1: Estimating New Equilibrium Point in WIP with Unknown Dynamics

As shown in Fig. 4, using a more accurate nonlinear dynamics model 5 showed its benefit in narrowing the *reality gap* in terms of position trajectories. Notably, on the graph below (pitch angle), only our method showed a similar pattern to the target graph (initially spiking upwards). Although two trajectories is not perfectly overlapped in ours case, results from our following experiments support that this is sufficient to enhance the performance of estimating the new equilibrium point (MSE less than $1.5rad$). The residual error might be caused by the other parametric errors (e.g., inertia of each link), coupled-dynamics, and non-parametric modeling error such as backlash and hysteresis.

Validation outcomes for the new equilibrium point estimation are summarized in Table I. These results demonstrate the efficacy of the data-driven estimator (e.g., LSTM), trained within a high-fidelity simulation from our framework, in narrowing the *reality gap* and thus improving new equilibrium point estimation accuracy. This underscores the importance of incorporating an elaborate dynamics model and optimizing its parameters for a seamless sim-to-real dynamics transfer. Notably, the MSE discrepancy between simulation (Sim) and real-world (Real) cases is a mere 0.001 rad.

Sim Param D-R counters *over-fitting*, improving estimation in simulation tests but performing poorly in real scenarios, as *Domain Randomization* doesn't directly bridge the model trajectory mismatch [30]. Generally, this can enhance model robustness by reducing sensitivity to non-essential features like noise, however, this attributes to increase dataset variance through parameter diversification, bringing the performance degradation in the regression problem. **Ours WO Opt** shows marginally better estimation over **Pure Sim**, suggesting the integration of a refined dynamics model (Eq. (5)) helps narrow the *reality gap*, despite randomized

parameters in training. **Sim Param Opt** shows the least accuracy, indicating that optimizing simulation parameters barely reduces the cost for ζ and hampers extracting vital features for new equilibrium point estimation, thus degrading estimation accuracy.

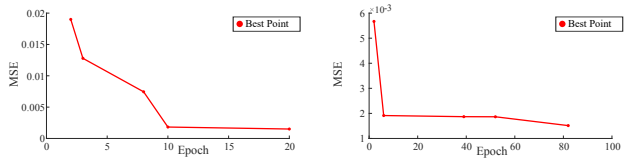
B. Experiment 2: Control Performance Validation

1) Balancing Task with Unknown Dynamics: In the balancing task with an unknown payload, utilizing a new equilibrium point effectively recover the WIP system's original position, as depicted in Fig. 6. More importantly, we observed that the WIP using the baseline LQR was prone to toppling over and continued to advance until reaching the mechanical limit of the rail. This shows that our method goes beyond simply increasing the tracking performance and contributes to improving the stability of the system. On the average considering all five cases, we observed 38% and 23% performance improvement in root mean square error of a position and angular velocity, respectively. The performance improvement in each case is as follows: [24.34%, 40.87%, 35.29%, 52.60%, 31.24%, 47.51%] in a position and [0.2574%, 0.3377%, 0.2837%, 0.3542%, 0.2759%, -0.1076%] in an angular position.

2) Tracking Task with Unknown Dynamics: In the tracking task, we used a predefined sinusoidal signal as a desired velocity ($\dot{xW}_{des} = 0.3sin(2\pi ft), f = 0.4$). Given that the baseline LQR fails at the operation's onset with an unknown payload attached to the system (see Fig. 6), we began the tracking task once the system stabilizes over time for a fair tracking task experiment. As shown in Fig. 7, using a new equilibrium point from our estimator led to the tracking performance improvement in both a position and angular position. On average, across all five cases, there was a 77 improvement in the root mean square error for position and a 60% improvement for angular velocity. The performance improvement in each case is as follows: [0.6646%, 0.7493%, 0.8588%, 0.5962%, 0.8811%, 0.9065%] in a position and [0.3407%, 0.5073%, 0.6740%, 0.6051%, 0.7142%, 0.7595%] in an angular position.

C. Ablation Study

1) Performance Comparison of Different Optimization Algorithm.: An additional experiment assessed the best optimization algorithm for identifying the nonlinear model parameters ζ, focusing on accuracy and time efficiency. Among seven algorithms tested using the SciPy package [27]—genetic algorithm (GA), particle swarm optimization (PSO), Nelder–Mead (NM), Powell, conjugate gradient (CG), BFGS, and sequential least squares programming (SLSQP)—only GA and PSO succeeded in parameter optimization, as shown in Fig. 8. This success likely stems from their global optimization capabilities, crucial for addressing the non-convex nature of our problem, whereas the others may stall at local minima. PSO outperformed GA in both convergence speed and accuracy, benefiting from its simpler velocity and position update processes. Optimizations were halted if exceeding 30 minutes, using default hyperparameters for each method.



(a) GA (b) PSO

Fig. 8: **Result of MSE according to update the parameter**. Graphs (a) and (b) compare the MSE (Eq. (7)) results of the GA and PSO, respectively, showing PSO's faster update rate allows it to achieve more epochs within the same timeframe.

TABLE II: **Performance benchmark for estimating the new equilibrium point between five different data-driven model.** Mean Square Error (MSE) of 146 test data obtained from a simulation (Sim) and 40 data obtained from the real world (Real). Length indicates the length of data trajectories which are an input of the data-driven estimator.

		LSTM		**Transformer**		**GRU**		**TCN**		**1D-CNN**	
		Sim	Real	Sim	Real	Sim	Real	Sim	Real	Sim	Real
Length	20	0.038	**0.036**	0.064	0.060	0.010	0.037	0.068	0.065	**0.009**	0.040
	40	0.031	0.042	0.064	0.060	0.008	0.036	0.087	0.065	**0.005**	**0.033**
	60	0.012	0.030	0.063	0.060	0.010	0.032	0.020	**0.023**	**0.005**	0.033
	80	0.024	**0.023**	0.028	0.024	0.020	0.030	0.038	0.027	**0.005**	0.035
Inference Time (sec)		0.035		**0.003**		0.02		**0.003**		0.006	

2) Performance Benchmark of Time-Series Data-Driven Model.: In Table II, we evaluate the new equilibrium point estimation performance using various time-series data-driven methods, also assessing the effect of sequence data length on accuracy. These methods were trained with high-fidelity simulations from our framework. Our results indicate improved performance with more historical data, suggesting initial WIP system movements provide insufficient features for identifying the new equilibrium point across different weights and CoM positions. TCN achieved the highest accuracy and fastest inference with shorter history data lengths. Conversely, LSTM excelled with 80 data points, potentially due to TCN's concurrent processing advantage and its structural benefits in addressing gradient issues common in recurrent networks. While 1D-CNN showed superior simulation data results, its real-world performance was less impressive. Notably, the choice of method did not critically affect our application's outcome.

D. Limitation

Although our method is more sample-efficient than model-free RL, it still requires a target domain dataset for Real-to-Sim adaptation. Collecting this dataset is challenging for high-dimensional systems with unknown payloads. Developing a Real-to-Sim adaptation method that does not rely on a target dataset is an interesting future direction. Additionally, to enhance simulation fidelity, errors from non-parametric models should be addressed using techniques like Gaussian Processes and Deep Neural Networks, which are commonly employed to represent such models.

VII. Conclusion

In this work, we present a framework designed to control a wheeled humanoid robot with unknown dynamics, aimed at safely and accurately delivering unknown objects. Ultimately, to facilitate system stability analysis and improve model accuracy, our approach is to estimate the new equilibrium of the system explicitly and utilize it in a model-based controller. Data-driven method is utilized to rapidly estimate the equilibrium point of the system with unknown dynamics. To be more efficient in the data collection process, we propose a novel real-to-sim adaptation that is capable of reducing the *reality gap* at the beginning of training a data-driven model and constructed dataset in the high-fidelity simulation. We conducted experiments with a physical inverted pendulum we developed as a simplified version of a wheeled humanoid. The results suggested that using a more accurate nonlinear dynamics with its optimized parameter showed benefit in narrowing the *reality gap*, contributing to improving the tracking performance of a model-based controller.

References

[1] Victor Klemm, Alessandro Morra, Lionel Gulich, Dominik Mannhart, David Rohr, Mina Kamel, Yvain de Viragh, and Roland Siegwart. Lqr-assisted whole-body control of a wheeled bipedal robot with kinematic loops. *IEEE Robotics and Automation Letters*, 5(2):3745–3752, 2020.

[2] Amartya Purushottam, Christopher Xu, Yeongtae Jung, and Joao Ramos. Dynamic mobile manipulation via whole-body bilateral teleoperation of a wheeled humanoid. *IEEE Robotics and Automation Letters*, 2023.

[3] Amartya Purushottam, Yeongtae Jung, Kevin Murphy, Donghoon Baek, and Joao Ramos. Hands-free telelocomotion of a wheeled humanoid. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8313–8320. IEEE, 2022.

[4] Boston Dynamics. Introducing handle. In *https://youtu.be/ - 7xvqQeoA8c*, 2017.

[5] DongHoon Baek, Yu-Chen Chang, and Joao Ramos. A study of shared-control with bilateral feedback for obstacle avoidance in whole-body telelocomotion of a wheeled humanoid. *IEEE Robotics and Automation Letters*, 2023.

[6] Marko Bjelonic, Ruben Grandia, Oliver Harley, Cla Galliard, Samuel Zimmermann, and Marco Hutter. Whole-body mpc and online gait sequence generation for wheeled-legged robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8388–8395. IEEE, 2021.

[7] Fabian Sonnleitner, Roberto Shu, and Ralph L Hollis. The mechanics and control of leaning to lift heavy objects with a dynamically stable mobile robot. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9264–9270. IEEE, 2019.

[8] Donghoon Baek, Amartya Purushottam, and Joao Ramos. Hybrid lmc: Hybrid learning and model-based control for wheeled humanoid robot via ensemble deep reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9347–9354. IEEE, 2022.

[9] Leilei Cui, Shuai Wang, Jingfan Zhang, Dongsheng Zhang, Jie Lai, Yu Zheng, Zhengyou Zhang, and Zhong-Ping Jiang. Learning-based balance control of wheel-legged robots. *IEEE Robotics and Automation Letters*, 6(4):7667–7674, 2021.

[10] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[11] Hua Chen, Bingheng Wang, Zejun Hong, Cong Shen, Patrick M Wensing, and Wei Zhang. Underactuated motion planning and control for jumping with wheeled-bipedal robots. *IEEE Robotics and Automation Letters*, 6(2):747–754, 2020.

[12] Songyan Xin and Sethu Vijayakumar. Online dynamic motion planning and control for wheeled biped robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3892–3899. IEEE, 2020.

[13] Maria Vittoria Minniti, Ruben Grandia, Kevin Fäh, Farbod Farshidian, and Marco Hutter. Model predictive robot-environment interaction control for mobile manipulation tasks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1651–1657. IEEE, 2021.

[14] Junheng Li and Quan Nguyen. Multi-contact mpc for dynamic loco-manipulation on humanoid robots. In *2023 American Control Conference (ACC)*, pages 1215–1220. IEEE, 2023.

[15] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[16] Ashish Kumar, Zhongyu Li, Jun Zeng, Deepak Pathak, Koushil Sreenath, and Jitendra Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168. IEEE, 2022.

[17] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[18] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[19] Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 6(2):911–918, 2021.

[20] Adam Allevato, Elaine Schaertl Short, Mitch Pryor, and Andrea Thomaz. Tunenet: One-shot residual tuning for system identification and sim-to-real robot task transfer. In *Conference on Robot Learning*, pages 445–455. PMLR, 2020.

[21] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[22] Yifeng Jiang, Tingnan Zhang, Daniel Ho, Yunfei Bai, C Karen Liu, Sergey Levine, and Jie Tan. Simgan: Hybrid simulator identification for domain adaptation via adversarial reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2884–2890. IEEE, 2021.

[23] Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv preprint arXiv:1906.01728*, 2019.

[24] Lawrence Perko. *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media, 2013.

[25] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[26] Jemin Hwangbo, Joonho Lee, and Marco Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018.

[27] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[28] Urdf - ros wiki. http://wiki.ros.org/urdf.

[29] Benjamin Katz, Jared Di Carlo, and Sangbae Kim. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6295–6301, 2019.

[30] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Robust and versatile bipedal jumping control through reinforcement learning. *arXiv preprint arXiv:2302.09450*, 2023.