Constrained Reinforcement Learning for Unstable Point-Feet Bipedal Locomotion Applied to the Bolt Robot

Constant Roux[1], Elliot Chane-Sane[1], Ludovic De Matteïs[1], Thomas Flayols[1],
Jérôme Manhes[1], Olivier Stasse[1,2] and Philippe Souères[1]

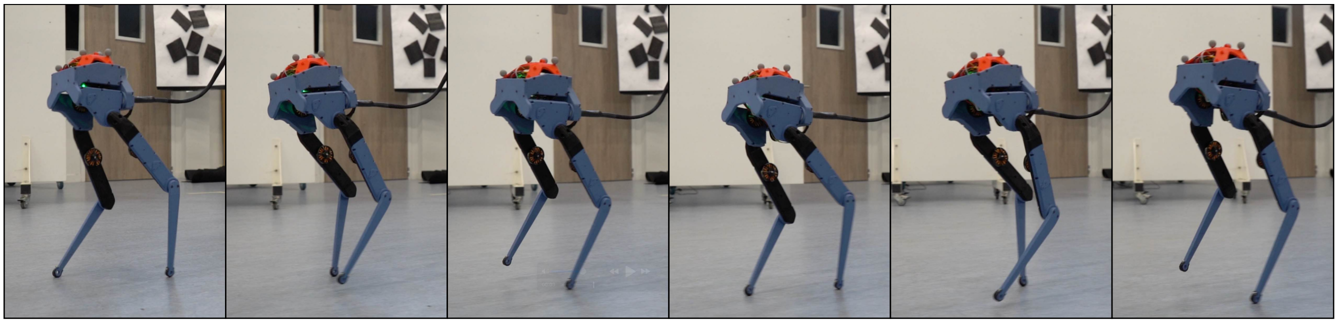

Fig. 1: The Bolt robot walking forward.

***Abstract*— Bipedal locomotion is a key challenge in robotics, particularly for robots like Bolt, which have a point-foot design. This study explores the control of such underactuated robots using constrained reinforcement learning, addressing their inherent instability, lack of arms, and limited foot actuation. We present a methodology that leverages Constraints-as-Terminations and domain randomization techniques to enable sim-to-real transfer. Through a series of qualitative and quantitative experiments, we evaluate our approach in terms of balance maintenance, velocity control, and responses to slip and push disturbances. Additionally, we analyze autonomy through metrics like the cost of transport and ground reaction force. Our method advances robust control strategies for point-foot bipedal robots, offering insights into broader locomotion. Videos and code are available at *https://gepetto.github.io/BoltLocomotion/*.**

I. Introduction

Traditionally, Model Predictive Control (MPC) has been the dominant method for controlling legged robots [1], [2]. However, the emergence of advanced Reinforcement Learning (RL) algorithms [3] and the development of GPU-accelerated simulators [4], [5] have driven a significant shift toward learning-based policies [6], especially in the context of quadrupedal locomotion [7], [8]. In recent years, such techniques have also demonstrated great success in controlling bipedal and humanoid robots [9], [10], [11], enabling these robots to perform in increasingly complex environments and often outperforming traditional MPC-based approaches.

Despite these advancements, sim-to-real transfer remains challenging for robots with unconventional morphological traits, such as limited ground contact or underactuated limbs [12], [13], [14], [15], [16]. Among these, bipedal robots with point feet [17], [18], [19], [20] pose a unique challenge due to their inherently unstable structure and minimal support area. While some prior works have achieved point-foot bipedal walking using quadrupedal robots [21], [22], these systems benefit from additional limbs that contribute to balance through angular momentum [23]—a strategy unavailable to arm-less bipedal robots. The scientific interest in point-foot bipedal robots is considerable.

As noted by Westervelt et al., "*a model of walking with a point contact is an integral part of an overall model of walking that is more anthropomorphic in nature than the current flat-footed walking paradigm*" [24]. Unlike rigid flat-footed robots—which often adopt overly simplified and non-anthropomorphic designs with stiff, unarticulated contact surfaces—point-foot systems better reflect the dynamics and control challenges seen in human locomotion. To explore this issue, we focus on the point-foot Bolt robot [25], a bipedal platform developed as part of the Open Dynamic Robot Initiative (ODRI). Bolt shares the same actuators as Solo, a quadrupedal robot from the same project, leveraging a modular and open-source design aimed at advancing torque-controlled legged locomotion. Notably, while Solo has been extensively studied [26], [27], [28], there has been relatively little research on Bolt [29], making it an interesting plateform for our subject of interest.

In this paper, we address the control of bipedal point-foot robots without arms, using the Bolt platform as a representative system. We analyze its dynamics and identify key challenges that must be tackled to enable robust locomotion. Our contributions are as follows:

- **First Application of Constrained RL on Point-Foot Bipedal Locomotion:** We apply constrained reinforcement learning to the problem of point-foot bipedal locomotion, demonstrating its feasibility on a setting that has received limited attention in prior work.

[1] LAAS-CNRS, Université de Toulouse, France, firstname.surname@laas.fr

[2] Artificial and Natural Intelligence Toulouse Institute, France, firstname.surname@laas.fr

- **Open-Source Training and Inference Pipeline:** We provide a fully open-source pipeline for training, inference, and logging on the low-cost, open-source Bolt robot, aiming to enhance reproducibility and support future benchmarking efforts.
- **Real-World Evaluation:** We conduct a real-world evaluation of the approach on the Bolt robot, providing insights into its performance and practical deployment.

The structure of the paper is as follows: Section II reviews related work on constrained RL and point-foot robots, Section III provides an in-depth description of the Bolt hardware, Section IV outlines the methodology, Section V presents experimental results, and Section VI concludes the paper.

II. Related Work

Recent advances in GPU-accelerated simulators [4], [5] have enhanced the efficiency of RL for robotic locomotion by enabling massively parallel training, which reduces training times [7]. RL has been successful in quadrupedal locomotion, with policies trained in simulation effectively transferring to real-world robots [30], [8], [27], [21]. However, RL for bipedal locomotion remains more challenging due to the inherent instability of bipedal morphology [9], [10], [11], especially in point-foot bipedal robots, where the underactuated stance phase complicates the control [22]. The absence of arms further increases the challenge by eliminating mechanisms to counteract angular momentum [18], [31], [32]. The Bolt robot, a point-foot biped, embodies many of the fundamental challenges in dynamic legged locomotion. Previous work has leveraged MPC and whole-body MPC to achieve stable walking behaviors [33], [29], [34]. In this work, we instead focus on RL to develop more robust and adaptable locomotion policies that can better handle uncertainty and disturbances.

Transferring RL policies to real-world bipedal robots is hindered by issues like unmodeled dynamics, sensor noise, and hardware limitations [9], [10], [11]. Constrained RL frameworks enhance safety and robustness by incorporating constraints, such as joint position and torque limits, as termination conditions [26], [35], [28]. One notable approach, *CaT (Constraints as Terminations)*, enforces these constraints during training to ensure hardware-compliant locomotion [26]. Alternative methods, such as [36], address similar challenges through different constraint-handling techniques. Furthermore, domain randomization techniques [12] expose policies to various training dynamics, such as variations in motor gains and friction coefficients, enhancing robustness to real-world discrepancies. Additionally, adaptive control strategies allow policies to adapt to real-world conditions by addressing discrepancies between simulation and reality [13], [14], [37]. In this work, we employ the *CaT* framework along with domain randomization to bridge the sim-to-real gap for the Bolt robot, allowing us to develop robust locomotion policies that account for real-world uncertainties.

Evaluation metrics for bipedal locomotion encompass multiple aspects of performance. Velocity tracking accuracy assesses how well the robot follows commanded velocity profiles, ensuring precise motion control [38]. Maximum achievable speed evaluates the robot's top velocity, reflecting its control limits [39], [40]. Autonomy is quantified through the cost of transport (CoT), which measures energy efficiency by calculating the energy consumed per unit distance traveled [41], [42]. Additionally, ground reaction force (GRF) profiles provide insights into balance, impact absorption, and adherence to physical constraints by analyzing force distribution during locomotion [43], [44]. Robustness is assessed through push recovery, which tests the robot's ability to regain stability after external perturbations [45], [46], [47], and slippage recovery, which evaluates its ability to maintain balance when encountering unexpected loss of traction [38], [47]. In this work, we comprehensively evaluate the Bolt robot's RL policies across all these metrics, analyzing its velocity tracking accuracy, maximum speed, energy efficiency, and robustness to external disturbances. This assessment offers insights into the behavior of the locomotion framework in real-world scenarios.

III. Hardware

A. Bolt Overview

Bolt (as shown in Fig. 1) is a bipedal robot developed by the ODRI [25]. It adopts a fully open-source, modular design philosophy aimed at enhancing accessibility and reproducibility in robotic research. Built with low-cost, off-the-shelf components and 3D-printed parts, Bolt features three torque-controlled degrees of freedom per leg and point feet. The robot omits arms entirely, which contributes to its lightweight form factor.

The Bolt robot relies on an external system, connected via a wired interface, for high-level computation and power supply. Commands are transmitted in real time via an Ethernet link to low-level motor controllers, enabling agile, closed-loop control while minimizing onboard complexity.

B. Locomotion Challenges

While Bolt's streamlined design facilitates experimentation, it introduces several critical challenges for dynamic locomotion:

- **Limited Stability:** Point feet provide no flat contact surface, significantly reducing the support polygon and making balance control more demanding, particularly during dynamic motions or on uneven terrain [19].
- **Angular Momentum Regulation:** The absence of arms limits the robot's ability to counterbalance body motion, placing a greater burden on the legs and torso to manage angular momentum during gait transitions and external disturbances [23].
- **Restricted Yaw Control:** Without an active yaw mechanism, Bolt struggles to reorient itself efficiently, which complicates tasks such as turning in confined spaces or navigating sharp trajectories [48].
- **Reduced Kinematic Redundancy:** Each leg's three actuators limit the robot's motion versatility, making agile behaviors such as running or adaptive stepping

more complex to achieve and requiring sophisticated control strategies.

IV. Method

In order to train a locomotion policy for the point-foot robot, Bolt, we use a sim-to-real approach. We first train the policy in simulation with deep RL and then directly transfer it to the real robot. This section describes the main components used to support learning and sim-to-real transfer on the Bolt robot.

A. Constrained Reinforcement Learning

Consider a Constrained Markov Decision Process defined as $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma, \mathcal{T}, \{\mathcal{C}^i\}_{i\in I})$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, respectively, γ is the discount factor, $\mathcal{R} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^+$ is the reward function, and $\mathcal{T} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}^+$ defines the probabilistic transition dynamics. The system is subject to constraints $\{\mathcal{C}^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}_{i\in I}$, where each constraint $\mathcal{C}^i$ yields a scalar penalty signal $c^i \in \mathbb{R}^+$ for a given state-action pair (s, a). We look for a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the discounted sum of future rewards:

$$\max_{\pi} \mathbb{E}_{\tau\sim\pi,\mathcal{T}} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) \right], \tag{1}$$

while ensuring the constraints satisfaction:

$$\mathbb{P}_{(s,a)\sim\rho_\gamma^\pi,\mathcal{T}} \left[c^i(s, a) > 0 \right] \leq \epsilon_i, \quad \forall i \in I. \tag{2}$$

We solve the problem defined by (1) and (2) using the *CaT* framework [26], built on Proximal Policy Optimization (PPO) [3], which incorporates constraints as termination conditions during training.

B. States and Actions

The state vector $s \in \mathcal{S}$ comprises the commanded base velocity $\begin{pmatrix}\mathbf{v}^{*\top} & \omega^*\end{pmatrix}^\top$, where $\mathbf{v}^*$ and ω^* represents respectively the desired linear and angular velocities in the robot's base frame. It also includes the base angular velocity ω, the projected gravity, the joint positions $\mathbf{q}$, the joint velocities $\dot{\mathbf{q}}$, and the previous actions. The six-dimensional action space corresponds to the angular position commands for the Bolt robot's joints. The policy outputs scaled delta actions, which are added to a predefined reference angular configuration and then tracked by a high-frequency proportional-derivative (PD) controller.

C. Rewards

The reward function is designed to promote precise tracking of the commanded base velocity, with separate terms for tracking the linear velocity in the xy-plane and the yaw velocity. The reward of the linear velocity is defined as:

$$R_{xy}(s) = \exp\left(-\frac{\left\|\mathbf{v}^* - \mathbf{v}\right\|^2}{\sigma_1^2}\right) \tag{3}$$

where $\mathbf{v}$ denote the measured linear velocity expressed in the robot's base frame, and σ_1 represents a scaling factor.

TABLE I: Constraints applied during training. A constraint is considered violated when its associated cost $c_i > 0$.

Safety Constraints	
Knee or base collision	$c_{\text{knee/base contact}} = \mathbb{1}_{\text{knee/base contact}}$
Upside-down state	$c_{\text{upsidedown}} = \mathbb{1}_{\text{upsidedown}}$
Foot contact force limit	$c_{\text{foot contact}_j} = \left\|f^{\text{foot}_j}\right\|_2 - f^{\text{lim}}$
Torque limits	$c_{\text{torque}_k} = \lvert\tau_k\rvert - \tau^{\text{lim}}$
Joint velocity limits	$c_{\text{joint velocity}_k} = \lvert\dot{q}_k\rvert - \dot{q}^{\text{lim}}$
Joint acceleration limits	$c_{\text{joint acceleration}_k} = \lvert\ddot{q}_k\rvert - \ddot{q}^{\text{lim}}$
Posture Constraints	
Base orientation	$c_{\text{ori}} = \lVert\text{base ori}_{\text{xy}}\rVert_2 - \text{base}^{\text{lim}}$
Hip orientation	$c_{\text{hip}_l} =\lvert \text{ hip ori}_l \rvert -\text{hip}^{\text{lim}}$
Knee orientation	$c_{\text{knee}_l} =\lvert \text{ knee ori}_l \rvert -\text{knee}^{\text{lim}}$
Gait Constraints	
Walking (single foot contact)	$c_{\text{foot contact}} = \lvert n_{\text{foot contact}} - 1\rvert$
Jumping (alternating zero or two foot contacts)	$c_{\text{foot contact}} = \lvert n_{\text{foot contact}} \bmod 2\rvert$
Foot air time	$c_{\text{air time}_j} = t^{\text{des}}_{\text{air time}} - t_{\text{air time}_j}$

Similarly, the reward for the yaw velocity is defined as:

$$R_{\text{yaw}} = \exp\left(-\frac{(\omega^* - \omega)^2}{\sigma_2^2}\right) \tag{4}$$

where σ_2 represents a scaling factor.

D. Constraints

To ensure safe and transferable locomotion, we define three categories of constraints during training: **safety constraints**, **posture constraints**, and **gait constraints**. These are detailed in Table I.

a) Safety Constraints: These constraints prevent critical failures that could damage the robot. They include prohibiting knee or base collisions with the ground, avoiding upside-down states, and limiting excessive foot contact forces f^{foot_j} of foot j with a threshold f^{lim}. Actuator safety is ensured by limiting joint torques τ_k, joint velocities $\dot{q}_k$, and accelerations $\ddot{q}_k$, where k denotes the actuator index. These are constrained within limits τ^{lim}, $\dot{q}^{\text{lim}}$, and $\ddot{q}^{\text{lim}}$, respectively.

b) Posture Constraints: Posture constraints regulate the robot's body configuration. The base orientation in the roll and pitch axes is constrained to prevent excessive tilting, with a limit defined by base^{lim}. Similarly, the orientations of the hip joints hip_l, and the knee joints knee_l (where l indicates the leg index), are bounded by limits hip^{lim} and knee^{lim}, ensuring a consistent posture during movement.

c) Gait Constraints: These constraints define the robot's locomotion gait. The walking constraint ensures that only one foot is in contact with the ground at any given time, expressed as $\lvert n_{\text{foot contact}} - 1\rvert$, where $n_{\text{foot contact}}$ denotes the number of feet in contact with the ground. For jumping, the constraint enforces that either zero or two feet are in contact with the ground, represented by $\lvert n_{\text{foot contact}} \bmod 2\rvert$. Foot air-time constraints ensure that each foot stays in the air for a sufficient amount of time during each locomotion cycle, defined by $t^{\text{des}}_{\text{air time}} - t_{\text{air time}_j}$, where $t_{\text{air time}_j}$ is the actual time that foot j is in the air.

Switching between walking and jumping behaviors requires retraining the network due to the mutually exclusive nature of the constraints.

E. Domain Randomization

To ensure robust zero-shot sim-to-real transfer, we employ domain randomization during training, as summarized in Table II. We introduce variations in both the environment and the robot's dynamics, training the policy to generalize better to real-world uncertainties. Additionally, noise is injected into the observations to simulate sensor imperfections and further improve robustness.

a) Environment and Dynamics Randomization: First, we randomize terrain properties by varying the ground friction coefficient and introducing uneven terrain profiles. The robot's dynamics is also randomized by modifying joint friction, shifting the center of mass (CoM) of the base, and perturbing the masses and inertia tensors of the robot's links. By randomizing the physics parameters, we prevent the policy from overfitting to both the simulator and the specific URDF model of the robot, ensuring it remains robust to real-world variations rather than exploiting simulator-specific artifacts.

b) State and Command Perturbations: To prevent overfitting to a fixed initial condition, the robot's initial state is randomized by applying random offsets to joint positions and velocities at spawn. External forces are randomly applied to the base to simulate real-world disturbances, and additional latency effects are introduced by simulating actuator delays and command variations.

c) Observation Noise: To account for sensor imperfections, noise is added to the observations. This noise injection further enhances robustness by preventing the policy from relying on overly precise state estimates.

F. Key Factors for Sim-to-Real Transfer

Among all randomization parameters, joint friction variations and actuator delays are particularly critical for successful sim-to-real transfer. Without accurately modeling joint friction, the robot fails to maintain balance and collapses immediately upon deployment. Likewise, actuator delays introduce inherent latency that must be accounted for to ensure smooth and stable motion execution. While other randomization parameters enhance overall robustness, these two factors are found to be indispensable for achieving reliable real-world locomotion.

V. Experiments

We conducted quantitative and qualitative experiments on the real robot to evaluate our controller, highlighting its strengths and identifying potential limitations.

A. Experimental Setup

The policies were trained using the IsaacLab framework [4], leveraging the PPO implementation from the CleanRL library [49], [3]. Our approach is implemented within our *CaT* framework [26], available on GitHub[1].

After successful training in simulation, the learned policies were directly deployed on the real Bolt bipedal robot, described in Sec. III. The policy runs at a frequency of 50Hz on an Apple Mac M3 Max CPU. Target joint positions are transmitted to the onboard PD controller, operating at a frequency of 10kHz. The complete inference pipeline, including code and logs, is available on GitHub[1] for reproducibility.

TABLE II: Domain randomization parameters for training. Key sim-to-real factors are in bold.

Category	Parameter	Distribution
Terrain	Ground friction coefficient	$\mathcal{U}(0.4,\ 1.5)$
	Height noise	$\mathcal{U}(0.0,\ 0.02)$ m, step = 0.005 m
Robot Dynamics	Mass scale factor	$\mathcal{U}(0.8,\ 1.2)$
	Inertia scale factor	$\mathcal{U}(0.8,\ 1.2)$
	Base CoM displacement	$\mathcal{U}(-0.02,\ 0.02)$ m
	Joint friction coefficient	$\mathcal{U}(0.01,\ 0.1)$
Initial State	Base position (x, y)	$\mathcal{U}(-0.5,\ 0.5)$ m
	Base yaw angle	$\mathcal{U}(-\pi,\ \pi)$ rad
	Base linear velocity	$\mathcal{U}(-0.3,\ 0.3)$ m/s
	Base angular velocity	$\mathcal{U}(-0.1,\ 0.1)$ rad/s
	Joint pos./vel. scale factor	$\mathcal{U}(0.9,\ 1.1)$
Events	**Actuation delay**	$\mathcal{U}(0,\ 2)$ simulation steps
	Push linear velocity (x, y)	$\mathcal{U}(-0.5,\ 0.5)$ m/s
	Push linear velocity (z)	$\mathcal{U}(-0.1,\ 0.1)$ m/s
	Push angular velocity	$\mathcal{U}(-0.5,\ 0.5)$ rad/s
Observation Noise	Base angular velocity	$\mathcal{U}(-0.2,\ 0.2)$ rad/s
	Projected gravity noise	$\mathcal{N}(0,\ 0.05^2)$ with bias $\mathcal{U}(0,\ 0.05)$
	Joint position noise	$\mathcal{U}(-0.01,\ 0.01)$ rad with bias $\mathcal{U}(0,\ 0.05)$ rad
	Joint velocity noise	$\mathcal{U}(-1.5,\ 1.5)$ rad/s

B. Evaluation Metrics

To quantitatively evaluate the performance of our approach, we introduce distinct evaluation metrics across three axes:

- **Performance Axis:** These metrics evaluate the robot's maximum velocity and the accuracy of its velocity control.
- **Autonomy Axis:** These metrics evaluate the short-term autonomy by monitoring power consumption, and long-term autonomy based on the ground reaction force computed for each foot.
- **Robustness Axis:** These metrics evaluate the controller's capacity to reject external disturbances.

Additionally, we conducted slip recovery and jumping experiments to further demonstrate the robustness and adaptability of the controller.

1) Performance Metrics:

a) Velocity Accuracy: To assess the accuracy of the velocity tracking of our controller, we computed the steady-state error for a set of velocity references along either the x-axis and the y-axis. The velocity tracking error $\bar{\boldsymbol{\epsilon}}$ is defined as the difference between the reference velocity $\mathbf{v}^*$ and the robot's mean steady-state velocity $\bar{\mathbf{v}}$.

b) Maximum Velocity: The maximum velocity is defined as the highest steady-state mean velocity attained along the x or the y-axis, in both directions, as the velocity reference is progressively increased. The process continues until the robot ceases to accelerate.

2) Autonomy Metrics:

a) Short-Term Autonomy: To assess the short-term autonomy of our controller, we measure the CoT for different

[1] *https://gepetto.github.io/BoltLocomotion/*

velocity references along the x and y-axes. The CoT is defined as:

$$\mathrm{CoT} = \frac{\sum_{i=0}^{N-1} \sum_{k=0}^{n_{\boldsymbol{u}}-1} P_{i,k}^{\text{loss}}}{Nmg \left\| \bar{\mathbf{v}} \right\|_2} \tag{5}$$

where N is the total number of time steps during the experiment, $n_{\boldsymbol{u}}$ is the number of joints, m is the robot's mass, g is the gravitational acceleration, and $\bar{\mathbf{v}}$ denotes the robot's average velocity throughout the experiment. The power loss $P_{i,k}^{\text{loss}}$ for actuator k at timestep i is:

$$P_{i,k}^{\text{loss}} = P_{i,k}^{J} + P_{i,k}^{f} \tag{6}$$

where $P_{i,k}^{J}$ and $P_{i,k}^{f}$ are the Joule and friction power losses, respectively, as described by Fadini et al. [50].

b) Long-Term Autonomy: To evaluate the potential damage caused by foot impacts on the ground, we estimate the ground reaction force using the following equation of motion:

$$M(\mathbf{q})\ddot{\mathbf{q}} + b(\mathbf{q}, \dot{\mathbf{q}}) = \tau(\mathbf{u}) + J_c(\mathbf{q})^T f \tag{7}$$

where $\mathbf{q}$ and $\dot{\mathbf{q}}$ represent the measured joint positions and velocities, respectively, τ denotes the joint torques generated by the control inputs $\mathbf{u}$, J_c is the contact Jacobian of the feet, $\ddot{\mathbf{q}}$ is the joint acceleration, and f is the ground reaction force.

This equation is solved using well-established algorithms from the literature [51], [52]. The computed ground reaction forces are then analyzed as part of our benchmark to assess impact-related wear and long-term autonomy.

3) Robustness Criteria:

a) Push Recovery: To quantify the robot's ability to reject external disturbances, an instrumented stick is used to apply controlled pushes to the robot's base. The pose of the stick and the force exerted on the robot are measured and the applied force is progressively increased in different directions until the robot loses stability and falls. The maximum impulse sustained before falling is recorded. The impulse is computed as the discrete integral:

$$\lambda = \delta t \sum_{i=1}^{M-1} \frac{f_i + f_{i-1}}{2} \tag{8}$$

where M denotes the total number of time steps during the push event, f_i represents the measured force at time step i, and δt is the time step duration.

C. Results

1) Performance Criteria:

a) Velocity Accuracy: Velocity commands range from -1 m/s to 1 m/s along the x and y-axes separately, with increments of 0.1 m/s. Each experiment is repeated at least three times per velocity reference. The steady-state velocity is determined by applying a moving average filter and extracting the stable segment of the signal. Results are presented in Fig. 3, where all velocities are normalized by $\sqrt{gL}$ (with L being the robot's leg length), corresponding to the Froude number normalization, as outlined in [53]. This normalization enables meaningful comparisons across different locomotion speeds and morphologies.

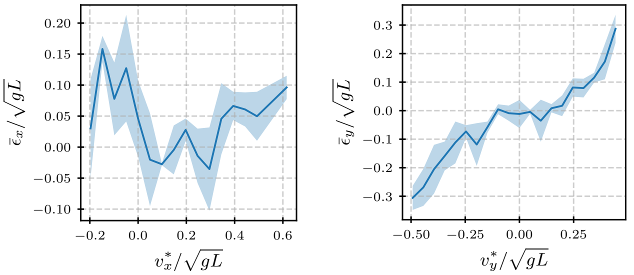

Fig. 3: Normalized steady-state velocity error versus normalized velocity command along the x-axis (left) and the y-axis (right). The solid line represents the mean, and the shaded area indicates the standard deviation.

TABLE III: Minimum and maximum normalized velocities achieved along the x and y axes.

	$v_{\text{min}}/\sqrt{gL}$	$v_{\text{max}}/\sqrt{gL}$
x-axis	-0.34	0.54
y-axis	-0.31	0.30

Overall, the robot successfully moves in the commanded direction (e.g., forward when instructed to do so). Along the x-axis, it achieves significantly higher forward velocities than backward ones but exhibits considerable drift, leading to poor velocity tracking. In contrast, along the y-axis, it demonstrates symmetric performance in both left and right directions. These results confirm that the robot is not only capable of balancing but also of following velocity commands. However, drift becomes more pronounced at higher velocity references. This drift can be attributed to the absence of linear velocity measurements in our setup, which eliminates direct feedback necessary for accurate reference tracking. As a result, the robot lacks immediate velocity feedback, making it harder to adjust movements in real-time for accurate control. This issue is compounded by the robot's inherently unstable morphology, further amplifying the challenge of maintaining precise control at higher speeds.

b) Maximum Velocity: Velocity references along the desired axis increase in increments of 0.1 m/s until the robot falls. The maximum steady-state velocity achieved is recorded. Table III summarizes the resulting performance, with all velocities normalized as before.

It is worth emphasizing that the robot achieves higher velocities in forward motion compared to backward, while exhibiting symmetric performance in lateral directions. When walking forward, the robot achieves a Froude number normalization of $Fr = 0.54$, which is comparable to the typical human walking value of $Fr \simeq 0.5$. Despite lower maximum velocities in the backward and lateral directions, the robot's performance remains competitive. Qualitative observations from video footage indicate that prior work [29] resulted in significantly lower velocities, highlighting the improved agility of our approach.

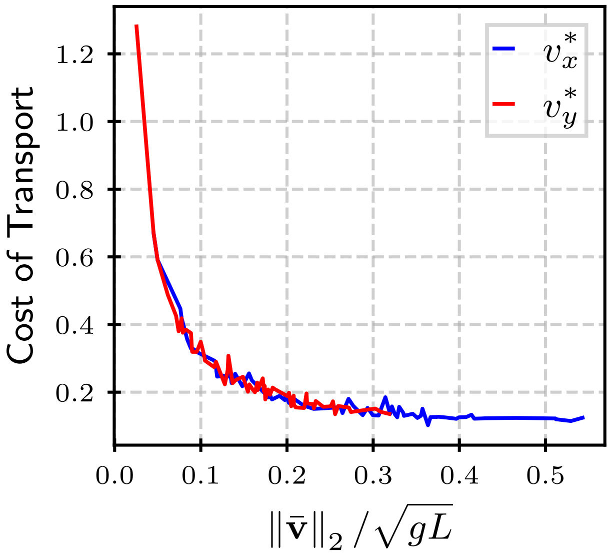

Fig. 4: Cost of transport as a function of the steady-state velocity norm. Blue represents velocity references along the x-axis, while red corresponds to those along the y-axis.

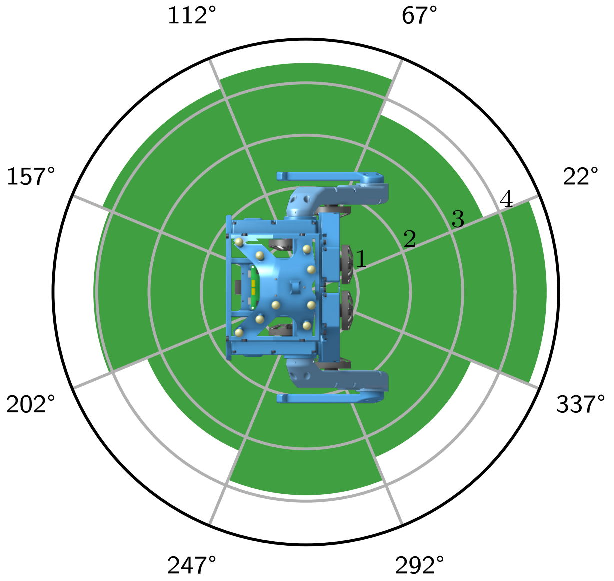

Fig. 5: Maximum impulse before falling (expressed as % of weight × seconds) as a function of the push angle, where 0° corresponds to a push applied to the back of the robot's base.

2) Autonomy Criteria:

a) Short-Term Autonomy: The CoT is evaluated at various steady-state velocities of the robot. Results are categorized based on the velocity command direction (forward/backward or left/right) and are presented in Fig. 4.

For velocity references along both the x and y-axes, the CoT decreases as the steady-state velocity norm increases. It can be used in future work to model battery consumption for an embedded version of the robot. Additionally, it establishes a useful benchmark for comparison in future studies.

b) Long-Term Autonomy: The mean ground reaction force (GRF) is found to be independent of the robot's velocity, remaining at $90\% \pm 6.2\%$ of the robot's weight. This value being below 100% is explained by the presence of short double support phases—periods during which both feet briefly share the load. Since the robot primarily walks with single-foot support, the average GRF per foot remains below full body weight. This indicates reduced GRF during gait transitions, which contributes to mechanical longevity. Furthermore, it confirms that the contact force constraint imposed during training is maintained in practice.

3) Robustness Criteria:

a) Push Recovery: The robot was subjected to over 300 pushes, covering a full range of directions around its base. Fig. 5 presents the maximum impulse, normalized as a percentage of the robot's weight, that the robot could withstand without falling, categorized in $45°$ quadrants.

Overall, the robot exhibits greater robustness to recover from pushes applied to the back and sides. Conversely, it is generally less robust when pushed from the front or at the corners, except for the upper right corner. This anomaly can be attributed to insufficient data points recorded in that region. For comparison, an impulse of 4% of the robot's weight over one second is equivalent to a 70 kg human being struck by a 2.8 kg ball over the course of 1 second, demonstrating the robustness of our method.

D. Additional Results

To qualitatively demonstrate the robustness of the balancing process, we present additional results. Fig. 6 shows the robot recovering from a slip after the carpet was pulled out while it was walking on it. These images are extracted directly from the accompanying video, which provides a more dynamic visualization of the entire recovery sequence.

Fig. 7 shows the robot performing jumps, enabled by modifying the gait constraint to allow ballistic phases (see Sec.IV-D). It successfully completed over 50 consecutive jumps, maintaining this behavior for more than 15 seconds, as demonstrated in the accompanying video.

VI. Conclusion

This work explored the control of a point-foot bipedal robot, emphasizing its significance in achieving anthropomorphic locomotion. Unlike flat-footed bipeds, point-foot robots face unique stability challenges, making their control an interesting research topic.

We developed a constrained RL approach and successfully transferred policies from simulation to the real robot without additional fine-tuning. Our experiments demonstrated that the robot can balance, track velocity commands, jump, and resist external disturbances, showcasing a high level of robustness.

Despite these achievements, real-world deployment remains challenging due to hardware fragility and external factors, such as the use of a wire connecting the robot

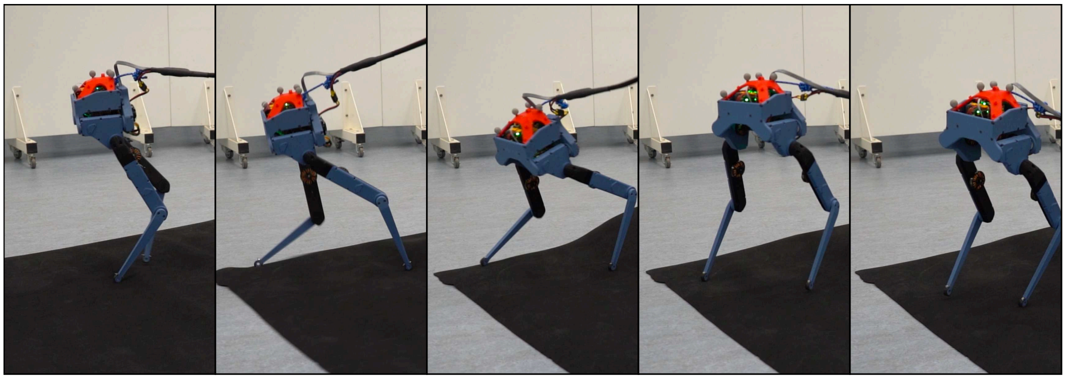

Fig. 6: Motion capture sequence of the robot recovering from a slip after the carpet was pulled away. Captured at 20 Hz.

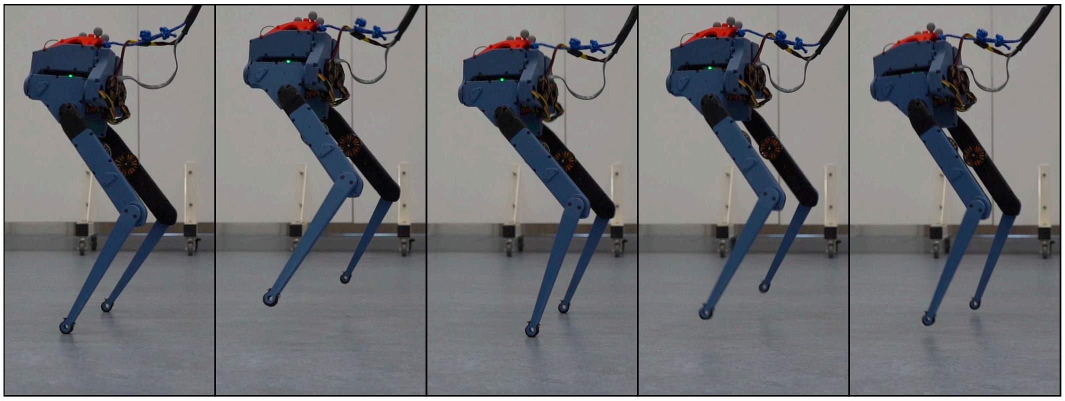

Fig. 7: Motion capture of the robot jumping. Captured at 33 Hz.

to an external computer. This setup, while necessary for communication and data processing, limits mobility and introduces additional complexities during operation. Future work will focus on refining robustness, exploring additional locomotion behaviors, and further improving real-world navigation capabilities.

AKNOWLEDGMENT

We thank the Fablaas and Electronics teams for their support in robot repairs, and the biomechanical team for constructing the force measurement stick. The Bolt platform was supported by ROBOTEX 2.0 (Grants ANR-10-EQPX-44-01, TIRREX-ANR-21-ESRE-0015) and AS2 (ANR-22-EXOD-0006), and ANR-19-PI3A-0004. This work was granted access to the HPC resources of IDRIS under the allocation 2025-AD011016104 made by GENCI.

REFERENCES

[1] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.

[2] H. Li and P. M. Wensing, "Cafe-mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control," *IEEE Transactions on Robotics*, vol. 41, pp. 837–856, 2025.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[4] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[5] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[6] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion: State of the art and future perspectives," *The International Journal of Robotics Research*, vol. 44, no. 8, pp. 1396–1427, 2025.

[7] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 91–100.

[8] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, p. 4630–4637, Apr. 2022.

[9] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," 2024.

[10] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," 2024.

[11] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," 2023.

[12] S. Chamorro, V. Klemm, M. de La Iglesia Valls, C. Pal, and R. Siegwart, "Reinforcement learning for blind stair climbing with legged and wheeled-legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8081–8087.

[13] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," 2021.

[14] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He,

N. Sobanbab, C. Pan, Z. Yi, G. Qu, K. Kitani, J. Hodgins, L. J. Fan, Y. Zhu, C. Liu, and G. Shi, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," 2025.

[15] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 12, p. 121006, 10 2022.

[16] H. Wang, H. Luo, W. Zhang, and H. Chen, "Cts: Concurrent teacher-student reinforcement learning for legged locomotion," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9191–9198, 2024.

[17] M. H. Raibert and E. R. Tello, "Legged robots that balance," *IEEE Expert*, vol. 1, no. 4, pp. 89–89, 1986.

[18] L. Dynamics, "Tron 1," Online, n.d., accessed: 10-Mar-2025. [Online]. Available: https://www.limxdynamics.com/en/tron1

[19] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-De-Wit, and J. Grizzle, "Rabbit: a testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[20] A. B. Ghansah, J. Kim, K. Li, and A. D. Ames, "Dynamic walking on highly underactuated point foot humanoids: Closing the loop between hzd and hlip," 2024. [Online]. Available: https://arxiv.org/abs/2406.13115

[21] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *arXiv preprint arXiv:2309.14341*, 2023.

[22] Y. Li, J. Li, W. Fu, and Y. Wu, "Learning agile bipedal motions on a quadrupedal robot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 9735–9742.

[23] Z. Gao, X. Chen, Z. Yu, L. Han, J. Zhang, and G. Huang, "Hybrid momentum compensation control by using arms for bipedal dynamic walking," *Biomimetics*, vol. 8, no. 1, p. 31, Jan. 2023.

[24] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, "Feedback control of dynamic bipedal robot locomotion," Oct. 2018.

[25] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wuthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, J. Fiene, A. Badri-Sprowitz, and L. Righetti, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, p. 3650–3657, Apr. 2020.

[26] E. Chane-Sane, P.-A. Leziart, T. Flayols, O. Stasse, P. Souères, and N. Mansard, "Cat: Constraints as terminations for legged locomotion reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[27] M. Aractingi, P.-A. Léziart, T. Flayols, J. Perez, T. Silander, and P. Souères, "Controlling the solo12 quadruped robot with deep reinforcement learning," *Scientific Reports*, vol. 13, no. 1, July 2023.

[28] E. Chane-Sane, C. Roux, O. Stasse, and N. Mansard, "Reinforcement learning from wild animal videos," 2024.

[29] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti, "Variable horizon mpc with swing foot dynamics for bipedal walking control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2349–2356, 2021.

[30] H. Zhang, S. Yang, and D. Wang, "A real-world quadrupedal locomotion benchmark for offline reinforcement learning," in *2024 International Joint Conference on Neural Networks (IJCNN)*, vol. 100. IEEE, June 2024, p. 1–7.

[31] T. Li, H. Geyer, C. G. Atkeson, and A. Rai, "Using deep reinforcement learning to learn high-level policies on the atrias biped," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 263–269.

[32] G. Mothish, K. Rajgopal, R. Kola, M. Tayal, and S. Kolathaya, "Stoch biro: Design and control of a low-cost bipedal robot," in *2024 10th International Conference on Control, Automation and Robotics (ICCAR)*, 2024, pp. 135–140.

[33] M. G. Boroujeni, E. Daneshman, L. Righetti, and M. Khadiv, "A unified framework for walking and running of bipedal robots," in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021, pp. 396–403.

[34] C. Roux, C. Perrot, and O. Stasse, "Whole-body mpc and sensitivity analysis of a real time foot step sequencer for a biped robot bolt," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, 2024, pp. 467–474.

[35] E. Chane-Sane, J. Amigo, T. Flayols, L. Righetti, and N. Mansard, "Soloparkour: Constrained reinforcement learning for visual locomotion from privileged experience," in *Conference on Robot Learning (CoRL)*, 2024.

[36] Y. Kim, H. Oh, J. Lee, J. Choi, G. Ji, M. Jung, D. Youm, and J. Hwangbo, "Not only rewards but also constraints: Applications on legged robot locomotion," *IEEE Transactions on Robotics*, vol. 40, pp. 2984–3003, 2024.

[37] N. Fey, G. B. Margolis, M. Peticco, and P. Agrawal, "Bridging the sim-to-real gap for athletic loco-manipulation," *arXiv preprint arXiv:2502.10894*, 2025.

[38] D. Youm, H. Jung, H. Kim, J. Hwangbo, H.-W. Park, and S. Ha, "Imitating and finetuning model predictive control for robust and symmetric quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7799–7806, 2023.

[39] I. S. Hiroshi Kimura and H. Miura, "Dynamics in the dynamic walk of a quadruped robot," *Advanced Robotics*, vol. 4, no. 3, pp. 283–301, 1989.

[40] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[41] M. Luneckas, T. Luneckas, J. Kriaučiūnas, D. Udris, D. Plonis, R. Damaševičius, and R. Maskeliūnas, "Hexapod robot gait switching for energy consumption and cost of transport management using heuristic algorithms," *Applied Sciences*, vol. 11, no. 3, p. 1339, Feb. 2021.

[42] O. Stasse, K. Giraud-Esclasse, E. Brousse, M. Naveau, R. Régnier, G. Avrin, and P. Souères, "Benchmarking the hrp-2 humanoid robot during locomotion," *Frontiers in Robotics and AI*, vol. 5, p. 122, 2018.

[43] W. Xu, R. Xiong, and J. Wu, "Force/torque-based compliance control for humanoid robot to compensate the landing impact force," in *2010 First International Conference on Networking and Distributed Computing*, 2010, pp. 336–340.

[44] D. Torricelli, J. Gonzalez-Vargas, J. F. Veneman, K. Mombaur, N. Tsagarakis, A. J. del Ama, A. Gil-Agudo, J. C. Moreno, and J. L. Pons, "Benchmarking bipedal locomotion: A unified scheme for humanoids, wearable robots, and humans," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 103–115, 2015.

[45] D. Ferigo, R. Camoriano, P. M. Viceconte, D. Calandriello, S. Traversaro, L. Rosasco, and D. Pucci, "On the emergence of whole-body strategies from humanoid robot push-recovery learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8561–8568, 2021.

[46] E. L. Dantec, W. Jallet, and J. Carpentier, "From centroidal to whole-body models for legged locomotion: a comparative analysis," Oct. 2024.

[47] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Walking control based on step timing adaptation," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 629–643, 2020.

[48] B. Ugurlu, J. Saglia, N. Tsagarakis, and D. Caldwell, "Yaw moment compensation for bipedal robots via intrinsic angular momentum constraint," *International Journal of Humanoid Robotics*, vol. 9, pp. 1–27, 12 2012.

[49] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. Araújo, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022.

[50] G. Fadini, T. Flayols, A. Del Prete, N. Mansard, and P. Souères, "Computational design of energy-efficient legged robots: Optimizing for size and actuators," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9898–9904.

[51] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[52] J. Carpentier, R. Budhiraja, and N. Mansard, "Proximal and sparse resolution of constrained dynamic equations," *Robotics: Science and Systems*, 2021.

[53] R. M. Alexander, *Principles of Animal Locomotion*, stu - student edition ed. Princeton University Press, 2003.