

Olaf: Bringing an Animated Character to Life in the Physical World

David Müller*, Espen Knoop*, Dario Mylonopoulos, Agon Serif,
Michael A. Hopkins, Ruben Grandia, and Moritz Bächer

Abstract—Animated characters often move in non-physical ways and have proportions that are far from a typical walking robot. This provides an ideal platform for innovation in both mechanical design and stylized motion control. In this paper, we bring Olaf to life in the physical world, relying on reinforcement learning guided by animation references for control. To create the illusion of Olaf’s feet moving along his body, we hide two asymmetric legs under a soft foam skirt. To fit actuators inside the character, we use spherical and planar linkages in the arms, mouth, and eyes. Because the walk cycle results in harsh contact sounds, we introduce additional rewards that noticeably reduce impact noise. The large head, driven by small actuators in the character’s slim neck, creates a risk of overheating, amplified by the costume. To keep actuators from overheating, we feed temperature values as additional inputs to policies, introducing new rewards to keep them within bounds. We validate the efficacy of our modeling in simulation and through experiments on the robot.

Index Terms—Reinforcement Learning; Machine Learning for Robot Control; Humanoid Robot Systems

I. INTRODUCTION

The field of legged robotics has traditionally been driven by goals centered on functionality, robustness, and efficiency [1]. This focus has enabled impressive dynamic capabilities, from hiking up mountains [2] to traversing challenging terrain [3], [4]. However, as robots enter domains that involve direct human participation, such as entertainment [5]–[8] and companionship [9], functional performance alone is no longer sufficient. In these settings, believability and character fidelity become central, shifting the engineering objective and imposing strict aesthetic constraints.

In this paper, we focus on bringing Olaf, an animated character, into the physical world (see Fig. 1). Unlike most robots, Olaf has a large, heavy head, small snowball feet, as well as a non-physical motion style. These characteristics impose strict constraints on both mechanical design and motion control, including the need to hide all mechanical components within tight spatial constraints. At the same time, even small inconsistencies, such as rough foot impacts or jitter, can break the character’s lifelike appearance. This sensitivity makes it particularly challenging to develop a convincing robotic representation of an animated character.

We address the mechanical challenges with a compact design that is fully hidden beneath a costume. To achieve the required mobility of the leg mechanism within the limited volume, we adopt an asymmetric leg design and conceal it under a soft foam skirt. This skirt also shapes the costume and creates the illusion of Olaf’s feet moving freely beneath his body. To preserve Olaf’s visual appearance, we rely on



Fig. 1: Olaf Robot.

small actuators and remote actuation through spherical and planar linkages. These are used throughout the character in the arms, mouth, and eyes.

The control of Olaf is based on Reinforcement Learning (RL), with imitation rewards centered around animation references. The combination of a large head and small actuators placed inside the slim costumed neck introduces a risk of overheating. To address this, we propose a policy that takes actuator temperatures as inputs. We incorporate a thermal actuator model into our simulation and formulate a reward that encourages thermal safety. Because harsh robotic footsteps break the character’s believability, we introduce a reward that noticeably reduces impact noise. Our reward formulation also includes joint limit constraints and a penalty for collisions between the feet. We complement the RL policies with classical control and system identification to actuate the mechanical linkages in the arms, eyes, and mouth.

Succinctly, our main contributions are:

- **Mechatronic design:** a compact, scale-accurate design of Olaf, featuring a novel asymmetric six-degrees-of-freedom (6-DoF) leg mechanism and the integration of remotely actuated spherical, planar, and spatial linkages to achieve high-fidelity, expressive motion in the arms, mouth, and eyes.
- **Thermal-aware policy:** a control policy that incorporates actuator temperature as an input and learns to prevent overheating through our proposed reward formulation.
- **Impact reduction reward:** a reward that substantially reduces footstep noise, helping to preserve the character’s believability.

*Joint first authors.

The authors are with Disney Research Imagineering.

II. RELATED WORK

Most legged robots take inspiration from Nature through their anthropomorphic [10]–[14] or zoomorphic [15]–[17] morphologies, and their design and control are predominantly guided by functional requirements [15], [18]. Recent advancements have pushed these boundaries through transformative mechanical designs, such as avian-inspired clutching linkages for energy efficiency [19] or multimodal skeletal structures that mimic small-scale mammals [20]. In this work, we instead focus on creating a robotic character, based on an artistic reference — bringing a well-known animated character into the physical world. The robot should freely walk and move like the animated character, with functional aspects such as energy efficiency and power density becoming secondary considerations.

Related work [7] created a *new* robotic character and presented a pipeline for animating and controlling it. Our work differs in that we are bringing an *existing* animated character to life, which requires navigating tradeoffs of functionality and believability within a tight design envelope. While the Cosmo robot [6] represents an existing character from a movie, we focus on a non-robotic, costumed character with less favorable proportions.

Bipedal and quadrupedal robots often have actuators at the joints [10], [16], or implement remote actuation through linkages [17], [21], [22]. Because the two legs of Olaf are hidden in its small main body, we depart from a traditional symmetric design, to maximize the workspace. We also use linkages to place actuators where there is space [23], [24], including spherical linkages at the shoulders.

For control, we rely on policies trained using RL that are conditioned on high-level control inputs. During runtime, these high-level control inputs are computed by a real-time character animation engine [7]. RL has led to tremendous progress in robust locomotion [25]–[28], imitation learning [29]–[32], and navigation of complex environments [33], [34]. More recently, RL has increasingly been used to account for additional real-world effects, ranging from energy losses in actuators [35] to impact-minimizing strategies that allow quieter locomotion [36], [37]. We build on these ideas to reduce Olaf’s stepping sound while preserving its characteristic gait, and further introduce thermal-aware policies with applications beyond our specific use case.

III. WORKFLOW

Our goal is to realize a robotic representation of Olaf, an animated character with characteristic, non-physical movements, and proportions atypical for robots. To meet these strict functional and creative requirements, the initial mechanical design began with the main backbone with actuated legs and neck (shown in green in Fig. 2). For animation authoring, we maintained an animation rig and animation references with the same degrees of freedom. To rapidly explore where to best place actuated degrees of freedom within the envelope of the character, we iteratively trained policies for standing and walking, evaluating the expressiveness of the character

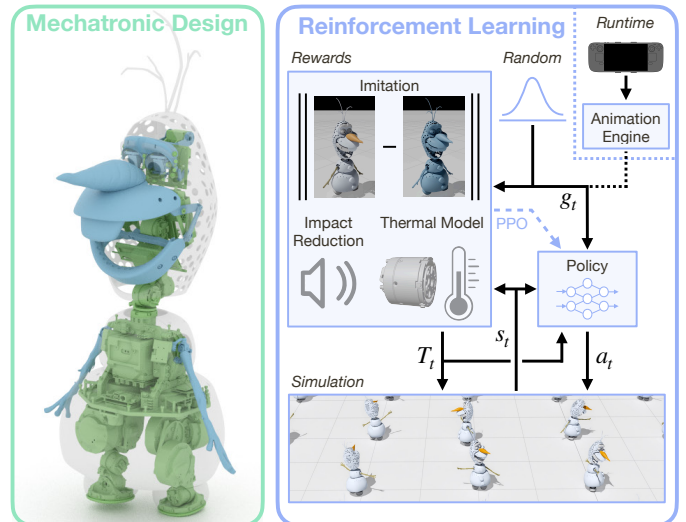


Fig. 2: **Mechatronic Design and RL-based Control.** We separate the **articulated backbone** from the **show functions**. The backbone is controlled via policies conditioned on the high-level control input g_t and trained using a combination of imitation, overheating, and impact rewards. During training, the control inputs are randomized, whereas at runtime, the Animation Engine generates control inputs from puppeteering commands.

skeleton in simulation. In a second phase, we added mechanical show functions—elements that drive expressive behavior such as the arms, mouth, eyes, and eyebrows (in blue) without affecting system dynamics. See Sec. IV for a description of the mechatronic design.

Control is divided into two layers: the articulation backbone, controlled by RL policies (Sec. V), and the show functions, using classical control methods (Sec. VI). The RL simulation model is derived from the mechanical design and augmented with actuator temperature dynamics. Policies are trained using a reward function that augments imitation terms for accurate tracking of kinematic references with penalties enforcing physical limits, such as joint ranges and actuator temperatures. We train separate walking and standing policies, each conditioned on control inputs g_t for animation tracking and interactive control.

At runtime (Sec. VII-C), Olaf is puppeteered via a remote interface. Following the design in [7], the *Animation Engine* processes these commands to switch policies, trigger animations and audio, and provide interactive joystick control.

IV. MECHATRONIC DESIGN

The presented robotic representation of Olaf stands 88.7 cm tall without hair and weighs 14.9 kg. It has 25 degrees of freedom in total; 6 per leg, 2 per shoulder, 3 in the neck, 1 in the jaw, 1 in the eyebrow, and 4 in the mechanical eyes. See Fig. 3 for an annotated cutaway view of the robot. We use Unitree and Dynamixel actuators as indicated in the figure, and on-board compute is provided by 3 computers. The form factor of the character presents several design challenges, discussed next.

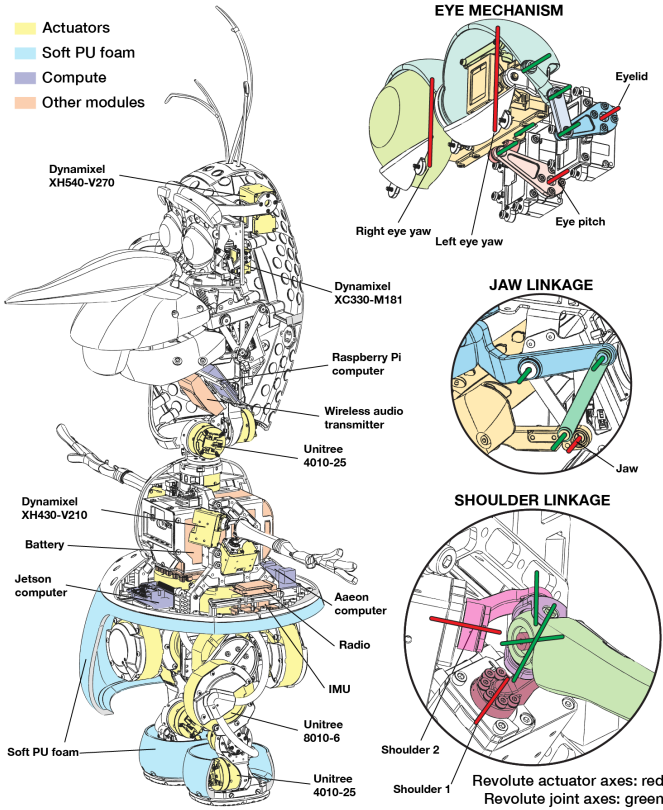


Fig. 3: **Mechatronic Design.** Shells and skirt have been cut away to show the interior. Note that the costume is not shown.

Compact Design Envelope: For the animated character, the two feet are free-floating snowballs beneath the body without visible legs. To emulate this, the robot design largely conceals the legs within the lower body, constraining the motion envelope of each leg to be within this bounding volume. To this end, we implement a novel 6-DoF leg design where one leg is inverted, such that the left leg has a rear-facing hip roll actuator and a forward knee, and the right leg has a forward hip roll actuator and a rear-facing knee (see Fig. 3). This configuration mitigates collisions between the two hip roll actuators and between the knees as the legs rotate in yaw. Furthermore, as the two legs are identical, instead of being mirrored, the part count is reduced.

The robot requires a 2-DoF shoulder; however, the limited space prevents placing the actuators at the joint. Instead, we place actuators within the torso and drive the shoulder through a spherical 5-bar linkage. See Fig. 3, and also the supplemental video. For the mouth, a single actuator drives both the upper and lower jaw. The lower jaw is actuated directly, and the upper jaw is coupled to this through a 4-bar linkage. The mechanical eyes have independent direct-drive eye yaw, along with eye pitch and eyelid, which are both remotely actuated through 4-bar linkages. All other joints are direct-drive.

Soft Shells: To conceal the legs within the lower snowball, while not overly restricting the range of motion, the lower snowball is designed as a flexible “skirt” made from polyurethane (PU) foam. This offers sufficient structure to maintain its shape while allowing deflection to accommodate

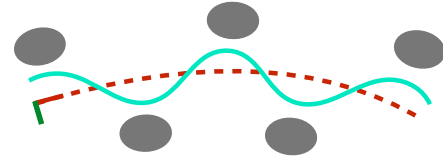


Fig. 4: **Path Frame.** Visualization of the path-frame concept from [7] and the robot’s center of mass.

larger leg movement, such as during recovery steps. The foot snowballs are constructed similarly. The flexible foam also absorbs impacts, reducing the severity of falls.

Costuming and Appendages: The costume is made of 4-way stretch fabric that can stretch both horizontally and vertically, allowing it to conform to the robot and its movements. A semi-rigid “boning” structure ensures that the costume maintains its shape over the cavity below the mouth. Around the eyes and the mouth, snap fasteners and magnets hold the costume in place. The arms, nose, buttons, eyebrows, and hair are held in place with magnets. This enables them to affix atop the costume (nose, buttons), and to break away in case of a fall or impact to mitigate damage.

V. REINFORCEMENT LEARNING

Building on previous work in character control [7], we use a separate walking and standing policy, each formulated as an independent RL problem tailored to its specific motion regime. At each time step, the agent produces an action \mathbf{a}_t according to a policy $\pi(\mathbf{a}_t | \mathbf{s}_t, \mathbf{g}_t)$, conditioned on the observable state \mathbf{s}_t and control input \mathbf{g}_t . The environment then returns the next state \mathbf{s}_{t+1} and a scalar reward $r_t = r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathbf{g}_t)$, which encourages accurate imitation of artist-defined kinematic motions while maintaining dynamic and robust balance.

To achieve invariance to the robot’s global pose and enable smooth transitions between policies, we use the path frame concept introduced in [7]. The path frame state at time t is

$$\mathbf{p}_t^{\text{PF}} := (x_t^{\text{PF}}, y_t^{\text{PF}}, \psi_t^{\text{PF}}), \quad (1)$$

where $(x_t^{\text{PF}}, y_t^{\text{PF}})$ denote the horizontal position and ψ_t^{PF} the yaw orientation.

During walking, the path frame advances by integrating the commanded path velocity \mathbf{v}_t^{PF} (see Fig. 4). During standing, the frame slowly converges toward the midpoint between the feet. Quantities expressed relative to this frame are denoted with the superscript \mathcal{P} . To prevent excessive deviation from the robot’s motion, the path frame is constrained to remain within a bounded distance of the torso.

A. Animation Reference

We start with animation references for walking and standing, created by artists using animation tools. These are conditioned on a control input \mathbf{g}_t , defined in (2). Notably, we use a gait generation tool [38] to design stylized walk cycles with heel-toe motion, capturing Olaf’s characteristic gait. We demonstrate the importance of this feature in Sec. VIII-A. Based on these references, the full kinematic target state \mathbf{x}_t is obtained through a generator function $f(\cdot)$ that maps the

path-frame state \mathbf{p}_t^{PF} and the policy-dependent control input \mathbf{g}_t to the kinematic target using interpolation and path-frame alignment. This mapping is expressed as

$$\begin{aligned} \mathbf{x}_t &:= (\mathbf{p}_t^{\text{P}}, \boldsymbol{\theta}_t^{\text{P}}, \mathbf{v}_t^{\text{P}}, \boldsymbol{\omega}_t^{\text{P}}, \mathbf{q}_t, \dot{\mathbf{q}}_t, c_t^{\text{L}}, c_t^{\text{R}}), \\ \mathbf{g}_t &:= \begin{cases} (\hat{\mathbf{q}}_t^{\text{neck}}, \hat{\boldsymbol{\theta}}_t, \hat{\mathbf{p}}_{z,t}) & \text{standing,} \\ (\hat{\mathbf{q}}_t^{\text{neck}}, \hat{\mathbf{v}}_t^{\text{PF}}) & \text{walking,} \end{cases} \\ \mathbf{x}_t &= \begin{cases} f(\mathbf{p}_t^{\text{PF}}, \mathbf{g}_t) & \text{standing,} \\ f(\mathbf{p}_t^{\text{PF}}, \mathbf{g}_t, \phi_t) & \text{walking,} \end{cases} \end{aligned} \quad (2)$$

where \mathbf{p}_t and $\boldsymbol{\theta}_t$ are the torso position and orientation (unit quaternion), \mathbf{v}_t and $\boldsymbol{\omega}_t$ are the linear and angular torso velocities, \mathbf{q}_t and $\dot{\mathbf{q}}_t$ are the joint positions and velocities, and c_t^{L} and c_t^{R} are the left and right foot contact indicators. Hats $\hat{\cdot}$ denote target quantities. For walking, $f(\cdot)$ additionally includes the gait phase variable ϕ_t . To ensure robustness and broad applicability, the control input \mathbf{g}_t is randomized across its full range during training.

B. Policy

Actions \mathbf{a}_t are position targets for Proportional-Derivative (PD) controllers at the joints. The robot’s proprioceptive state is

$$\mathbf{s}_t := (\mathbf{p}_t^{\text{P}}, \boldsymbol{\theta}_t^{\text{P}}, \mathbf{v}_t^{\text{R}}, \boldsymbol{\omega}_t^{\text{R}}, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, \mathbf{a}_{t-2}, \mathbf{T}_t, \phi_t), \quad (3)$$

where \mathbf{p}_t^{P} and $\boldsymbol{\theta}_t^{\text{P}}$ form the root pose relative to the path frame, and \mathbf{v}_t^{R} and $\boldsymbol{\omega}_t^{\text{R}}$ denote the torso velocities expressed in the root frame. We also append the joint positions \mathbf{q}_t , joint velocities $\dot{\mathbf{q}}_t$, the actions of the two previous time steps, \mathbf{a}_{t-1} and \mathbf{a}_{t-2} , and the temperature \mathbf{T}_t of the actuators. For walking, the policy is additionally conditioned on the gait phase variable ϕ_t .

C. Reward Formulation

The reward includes four components: *imitation*, *regularization*, *limits*, and *impact reduction*,

$$r_t = r_t^{\text{imitation}} + r_t^{\text{regularization}} + r_t^{\text{limits}} + r_t^{\text{impact reduction}}. \quad (4)$$

The *imitation* and *regularization* terms follow standard practice [7], [29] and encourage accurate tracking of the reference motion with action penalties. The *limit* terms capture constraints arising from Olaf’s compact mechanical design. The *impact reduction* term reduces foot impacts and thereby significantly lowers footstep noise.

A detailed breakdown of the reward terms is provided in Tab. I with reward weights reported in Tab. III, where hats $\hat{\cdot}$ denote target quantities derived from \mathbf{x}_t . The time index t is omitted for readability. Below, we describe the reward functions in more detail.

The reward $r_t^{\text{imitation}}$ incentivizes accurate imitation of the kinematic reference motion. Joint weights differ between the neck and legs to account for their substantially different reflected inertia. We apply early termination when the head, torso, upper legs, or arms are in contact with the ground. We additionally add regularization terms $r_t^{\text{regularization}}$ to penalize

TABLE I: **Reward Terms.** Hats $\hat{\cdot}$ denote reference quantities. Control-barrier rewards enforce thermal and joint limits, while penalizing large foot velocity rates reduces impact noise. \boxminus is the $SO(3)$ log-map orientation difference and $\mathbb{1}[\cdot]$ is the indicator function.

Name	Reward Term
<i>Imitation</i>	
Torso position xy	$\exp(-200.0 \cdot \ \mathbf{p}_{x,y} - \hat{\mathbf{p}}_{x,y}\ _2^2)$
Torso orientation	$\exp(-20.0 \cdot \ \boldsymbol{\theta} \boxminus \hat{\boldsymbol{\theta}}\ _2^2)$
Linear vel. xy	$\exp(-8.0 \cdot \ \mathbf{v}_{x,y} - \hat{\mathbf{v}}_{x,y}\ _2^2)$
Linear vel. z	$\exp(-8.0 \cdot \ \mathbf{v}_z - \hat{\mathbf{v}}_z\ _2^2)$
Angular vel. xy	$\exp(-2.0 \cdot \ \boldsymbol{\omega}_{x,y} - \hat{\boldsymbol{\omega}}_{x,y}\ _2^2)$
Angular vel. z	$\exp(-2.0 \cdot \ \boldsymbol{\omega}_z - \hat{\boldsymbol{\omega}}_z\ _2^2)$
Leg joint pos.	$-\ \mathbf{q}_l - \hat{\mathbf{q}}_l\ _2^2$
Neck joint pos.	$-\ \mathbf{q}_n - \hat{\mathbf{q}}_n\ _2^2$
Leg joint vel.	$-\ \dot{\mathbf{q}}_l - \hat{\dot{\mathbf{q}}}_l\ _2^2$
Neck joint vel.	$-\ \dot{\mathbf{q}}_n - \hat{\dot{\mathbf{q}}}_n\ _2^2$
Contact	$\sum_{i \in \{L, R\}} \mathbb{1}[c_i = \hat{c}_i]$
Survival	1.0
<i>Regularization</i>	
Joint torques	$-\ \boldsymbol{\tau}\ _2^2$
Joint acc.	$-\ \ddot{\mathbf{q}}\ _2^2$
Leg action rate	$-\ \mathbf{a}_l - \mathbf{a}_{t-1,l}\ _2^2$
Neck action rate	$-\ \mathbf{a}_n - \mathbf{a}_{t-1,n}\ _2^2$
Leg action acc.	$-\ \mathbf{a}_l - 2\mathbf{a}_{t-1,l} + \mathbf{a}_{t-2,l}\ _2^2$
Neck action acc.	$-\ \mathbf{a}_n - 2\mathbf{a}_{t-1,n} + \mathbf{a}_{t-2,n}\ _2^2$
<i>Limits</i>	
Neck temperature	$-\ \min(-\hat{\mathbf{T}}_n + \gamma_T(\mathbf{T}_{\max} - \mathbf{T}_n), \mathbf{0})\ _1$
Joint limits (lower)	$-\ \min(\hat{\mathbf{q}} + \gamma_q(\mathbf{q} - (\mathbf{q}_{\min} + \mathbf{q}_m)), \mathbf{0})\ _1$
Joint limits (upper)	$-\ \min(-\hat{\mathbf{q}} + \gamma_q((\mathbf{q}_{\max} - \mathbf{q}_m) - \mathbf{q}), \mathbf{0})\ _1$
Foot-Foot collision	$-\mathbb{1}[\text{contact}(L, R)]$
<i>Impact Reduction</i>	
Sound suppression	$-(\sum_{i \in \{L, R\}} \min(\Delta v_{i,z}^2, \Delta v_{\max}^2))$

excessive joint torques and encourage smooth actions, thereby reducing vibrations and unnecessary effort.

The reward r_t^{limits} penalizes violations of critical physical constraints. We incorporate a temperature reward term as detailed in Sec. V-D, and implement joint limit penalties, described in Sec. V-E. Finally, to prevent self-collisions between Olaf’s two snowball feet, we include a penalty for foot-foot contact.

The term $r_t^{\text{impact reduction}}$ penalizes changes in velocity along the gravity direction between simulation steps, thereby producing smoother foot motions and reducing impact noise. Because the physics engine can generate large velocity changes during contact resolution, which may result in a large reward term and destabilize critic learning, we saturate the impact-reduction reward term in the reward computation.

D. Thermal Modeling

Olaf’s slim, costume-covered neck requires small actuators to support a heavy head, which caused frequent overheating in early experiments. We address this by requiring the actuator temperature T to stay below a maximum temperature T_{\max} , as formalized through the inequality constraint in (5a). We transform this constraint, which depends on the slowly varying

temperature state, into a Control Barrier Function (CBF, [39]) condition (5c). The CBF condition shapes the desired behavior locally in time by imposing constraints on the time derivative. Intuitively, it ensures that as the temperature approaches or exceeds the maximum, the system responds by maintaining $\dot{T} \leq 0$, thereby preventing overheating

$$h_T(T) = T_{\max} - T \geq 0, \quad (5a)$$

$$\dot{h}_T(T) + \gamma_T h_T(T) \geq 0, \quad \gamma_T > 0, \quad (5b)$$

$$-\dot{T} + \gamma_T (T_{\max} - T) \geq 0, \quad \gamma_T > 0. \quad (5c)$$

The set of CBF constraints per actuator is translated into a penalty by computing the total violation, as defined in Tab. I. To implement the thermal CBF in simulation, we require a model of the actuator thermal dynamics. These dynamics are dominated by electrical Joule heating P , which scales with τ^2 since torque $\tau \propto I$ and $P \propto I^2$. We therefore model the temperature as a first-order system driven by squared torque

$$\dot{T} = -\alpha(T - T_{\text{ambient}}) + \beta\tau^2, \quad (6)$$

neglecting mechanical heat generation. The parameters α , β , and T_{ambient} are fitted from data, as detailed in Sec. VII-A.

E. Joint Limits

To prevent joint-limit violations, we use a similar reward function based on CBF conditions (8), which enforces a margin q_m from each joint's physical limits q_{\min} and q_{\max} . For each joint, we define

$$h_q(q) = \begin{cases} q - (q_{\min} + q_m) \geq 0, & \text{lower limit,} \\ (q_{\max} - q_m) - q \geq 0, & \text{upper limit,} \end{cases} \quad (7)$$

with the corresponding per-joint CBF constraints

$$\begin{aligned} \dot{q} + \gamma_q (q - (q_{\min} + q_m)) &\geq 0, & \text{lower limit,} \\ -\dot{q} + \gamma_q ((q_{\max} - q_m) - q) &\geq 0, & \text{upper limit,} \end{aligned} \quad (8)$$

where $\gamma_q > 0$. We choose a margin of $q_m = 0.1$ rad and set $\gamma_q = 20$.

VI. SHOW FUNCTIONS

Olaf's show functions — the eyes and eyebrows, the jaw mechanism, and the arms — have low inertia and therefore minimally affect system dynamics. For this reason, they are separated from the main articulated backbone and controlled using classical methods. To control Olaf's show functions, we must map their functional space, which is the space in which motions are animated and composited, to actuator space. This mapping is derived using a forward-kinematics solver [40] by uniformly sampling the functional region of interest and fitting a polynomial.

For Olaf's eyes, the functional space includes left and right eye yaw, coupled eye pitch, and eyelid closure. For the given eye mechanism, a first-order polynomial per actuator is sufficiently accurate. Olaf's left and right arms are implemented as spherical 5-bar linkages, each driven by two actuators. We parameterize their functional coordinates using two serial revolute angles: *arm swing* followed by *arm pitch*. Arm swing maps directly to the first actuator, whereas arm

TABLE II: **Thermal Model and Reward Parameters.** Neck actuator thermal model coefficients and reward parameters, specifying temperature dynamics, allowed limits, and the control-barrier coefficient.

Thermal Model		Reward Function	
α	0.038	T_{\max}	80 °C
β	0.377	T_{clip}	[70 °C, 85 °C]
T_{ambient}	43.94	γ_T	0.312

pitch is coupled through both actuators. The second actuator position is obtained through a cubic polynomial fit. After applying the actuator mapping, all eye and arm actuators are controlled using a PD loop.

For the jaw, the costume introduces significant external forces through fabric tension when the mouth is closed and wrinkling when it opens, which degrade tracking performance. To compensate for these effects, we add a feedforward term $\tau_{\text{ff}}^{\text{jaw}}(q^{\text{jaw}})$ to the PD controller. We estimate this feedforward term by measuring the torque required to hold a set of uniformly sampled jaw angles q^{jaw} across the full range of motion. Using least squares, we fit a first-order polynomial with an additional cosine term that captures the non-linearity observed in the data

$$\tau_{\text{ff}}^{\text{jaw}}(q^{\text{jaw}}) = c_0 + c_1 q^{\text{jaw}} + c_{\cos} \cos(q^{\text{jaw}}), \quad (9)$$

where c_0 , c_1 , and c_{\cos} are the fitted model parameters.

VII. IMPLEMENTATION DETAILS

A. Thermal Model and Reward

We fit the parameters of the thermal model (α , β , T_{ambient}) in (6) using a least-squares regression applied to an explicit Euler discretization of the thermal dynamics, sampled at 50 Hz to match the policy rate. The regression is performed on 20 min of recorded data. A quantitative evaluation of the resulting model is provided in Sec. VIII-B. All parameters used in the thermal reward model are summarized in Tab. II. The maximum actuator temperature, T_{\max} , is set to 80 °C, a value determined experimentally to prevent overheating while maintaining a conservative safety margin.

To limit the temperature range the policy needs to be exposed to during training, we clip the temperatures used in policy observations and rewards to the interval $[T_{\text{clip,min}}, T_{\text{clip,max}}]$. To ensure that the constraint in (5c) remains feasible at the upper clipping boundary, we choose γ_T such that the constraint is satisfied under the smallest possible heat generation, i.e., when $\tau^2 = 0$.

B. Training

We train the policies using Proximal Policy Optimization (PPO) [41]. The critic receives privileged information, including noiseless measurements, friction parameters, and terrain height samples. We add noise to the actor observations and apply randomized disturbance forces during training. Both actor and critic are implemented as three-layer MLPs with 512 units per layer. All reward weights are listed in Tab. III.

TABLE III: **Reward Weights.** Weights used for the standing and walking policies. Two values indicate *Standing / Walking*; a single value applies to both.

Reward Name	Standing /Walking	Reward Name	Standing /Walking
Torso position xy	1.0/4.0	Neck action rate	5.0/10.0
Torso orientation	2.0/1.5	Leg action rate	2.0/5.0
Linear vel. xy	1.5/2.5	Leg action acc.	0.5/1.0
Linear vel. z	1.0	Neck action acc.	15.0/10.0
Angular vel. z	1.5	Neck temperature	2.0
Leg joint pos.	15.0	Joint limits	0.5/0.2
Neck joint pos.	40.0	Foot-Foot collision	10.0
Leg joint vel.	$1.0 \cdot 10^{-3}$	Impact reduction	$2.5 \cdot 10^{-3}$
Neck joint vel.	0.5	Joint torques	$1.0 \cdot 10^{-3}$
Contact	2.0/1.0	Joint acc.	$2.5 \cdot 10^{-6}$
Survival	20.0		

Training is performed in Isaac Sim, running 8192 environments in parallel on a single RTX 4090 GPU. Policies are trained for 100k iterations (approximately 2 d).

C. Runtime

After training, the policy networks are frozen and deployed to the robot’s on-board computer. The proprioceptive state s_t is estimated using a state estimator that fuses IMU and actuator measurements [42]. The policy runs at 50 Hz, and its output is upsampled to the 600 Hz actuator rate using a first-order hold (i.e., linear interpolation between successive actions). Finally, the upsampled actions are passed through a low-pass filter with a cut-off of 37.5 Hz to ensure smooth motor commands.

Using the architecture from [7], puppeteering commands are processed by an *Animation Engine*, which blends triggered content and maps real-time puppeteering inputs to policy control input g_t , show function signals, and audio triggers through a three-stage process:

- **Background animations:** plays looped whole-body animations to introduce subtle idle behaviors such as eye saccades and arm adjustments.
- **Triggered animations:** layers short animated clips (e.g., gestures or spoken lines) on top of the background animations.
- **Joystick-driven control:** adjusts the output pose based on teleoperation inputs. During standing, it controls gaze and body posture; during walking, it controls gaze and path velocity.

VIII. RESULTS

We evaluate Olaf’s mechatronic design in the real world, with its performance demonstrated directly in the visual results. The asymmetric leg design enables faithful imitation of Olaf’s characteristic animations while adhering to all design requirements. The magnetic arms and nose enable in-character gags, as highlighted in the supporting video.

In the following, we analyze the robot’s control mechanisms in detail.

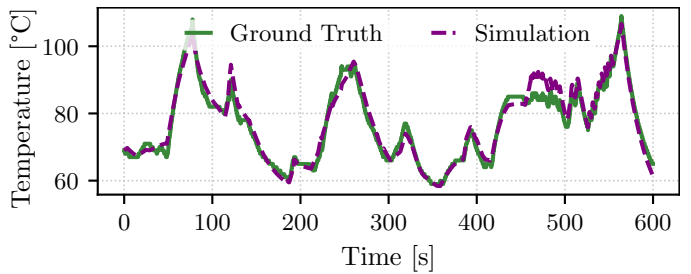


Fig. 5: **Thermal Model.** Validation of the thermal model in (6), comparing **predicted** actuator temperatures, T , with **measured** values over a 10 min rollout. Temperatures are quantized as reported by the actuator.

A. Tracking Performance

We compare the kinematic reference to the executed motion on the robot (see supplementary video). To quantify this, we evaluate the mean absolute joint-tracking error over the full trajectory and all joints of our walking and standing policies across the full range of control inputs g_t for 5 min each. For the standing policy, we achieve an error of $3.87^\circ \pm 2.40^\circ$, and for the walking policy, $4.02^\circ \pm 2.01^\circ$.

We additionally analyze the effect of Olaf’s characteristic heel-toe walking by training a policy without it to assess its effect on the visual appearance of his gait. As shown in the video, the resulting motion appears more robotic.

B. Thermal Model Evaluation

As shown in Fig. 5, we evaluate the thermal model fitted with the parameters in Tab. II by simulating the temperature evolution over an unseen 10 min trajectory, initialized to the measured value at time 0. Across the full trajectory, the model achieves a mean absolute error of 1.87°C , demonstrating good predictive accuracy.

The timescale of the thermal model is far slower than that of the balancing dynamics, making it a challenging RL objective. To evaluate the effectiveness of the proposed thermal reward, we compare the temperature evolution of the neck-pitch actuator, identified in our experiments as the actuator most prone to overheating, and the mean absolute joint-tracking error for policies trained with and without the thermal reward. The evaluation is performed on a trajectory where Olaf looks upward and follows a predefined motion, slowly turning his head left and right.

As shown in Fig. 6, the baseline policy without the thermal reward causes the actuator temperature to rise rapidly, reaching 100°C within 40 s, at which point the experiment was stopped to prevent actuator damage. The policy with the thermal reward exhibits only a slightly larger tracking error, while the actuator temperature rises significantly slower. This is explained by the squared torque, which shows that the policy reduces torque usage well before reaching the temperature limit, while maintaining nearly the same tracking accuracy. As the temperature approaches the 80°C threshold, the squared torque increases as the policy adjusts the head toward a more horizontal orientation. This requires more torque initially, but

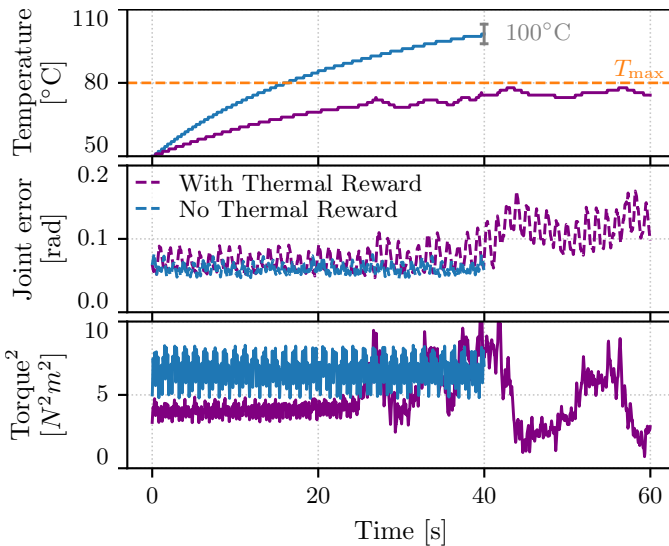


Fig. 6: **Thermal Reward Evaluation.** Neck-pitch temperature, mean absolute joint-tracking error, and squared neck-pitch torque for policies trained **with** and **without** the thermal reward. The thermal reward slows temperature rise while maintaining tracking at low temperatures and slightly relaxes tracking near the temperature limit to prevent overheating.

lowers torque demand over time. To evaluate long-duration performance, we executed the same motion continuously for 1 h, during which the system reached a thermal equilibrium. Over the final minute of the experiment, we measured a mean actuator temperature of 77.3°C and a mean absolute joint tracking error of 0.14 rad. The video further illustrates this behavior, showing how the policy gradually relaxes tracking near the temperature limit.

C. Foot Impact

We evaluate the foot impact reduction reward for sound suppression by comparing its addition to the baseline on hardware. Over a 5 min run, this reward reduces the mean sound level by 13.5 dB, as clearly noticeable in the video. Despite the reward, tracking performance and similarity to the kinematic reference are largely preserved.

Fig. 7 shows the vertical foot velocity and position profiles for the reference, as well as policies trained with and without the foot impact reduction reward during a swing phase. The reward acts as a regularizer: the overall trajectory is preserved, but small nuances of the reference, such as the mid-swing dip of the foot, are smoothed out. The no-impact-reduction policy imposes less smoothing, but exhibits higher peak velocities at foot impact. This demonstrates that the reward effectively reduces impact forces while maintaining the overall motion profile.

D. System Deployment

We have validated the robot through extended deployments and real-world testing. The thermal reward prevented actuator overheating, with no noticeable detrimental effect on the performance.

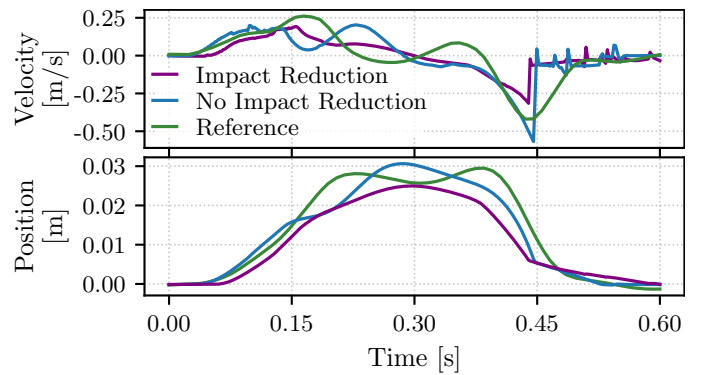


Fig. 7: **Foot Impact Reduction.** Comparison of z-foot velocity and position for the **reference** and policies trained **with** and **without** the foot impact reduction reward.

IX. CONCLUSION

This work presented Olaf, a freely walking robot that accurately imitates the animated character, which is challenging to physically represent, in terms of style and appearance. We hope this work inspires others to push the boundaries beyond standard bipedal and quadrupedal robotics.

We addressed challenging design requirements by proposing an asymmetric 6-DoF leg mechanism hidden beneath a foam skirt and the integration of remotely-actuated linkages. We tackled control requirements by using RL and impact-reducing rewards to significantly reduce stepping sound. We further incorporated control barrier function constraints to mitigate actuator overheating with a thermal model and to prevent joint-limit violations.

Although the results are specific to Olaf, we expect our impact-reducing rewards to decrease wear and extend a robot’s operational lifetime. In our experiments, the CBF reward formulation has prevented constraint violations, providing an effective tool for handling constraints within an RL context. CBF rewards are particularly useful for long-horizon effects such as thermal management, enabling a robot to self-manage temperature via learned policies.

Several open research directions remain. A higher fidelity thermal model could capture contributions of mechanical effects like friction or gradual heating of actuator enclosures during extended operation. In this work, interaction forces between the costume and legs were handled through domain-randomized disturbance forces. Explicitly modeling these effects could reduce reliance on randomization and provide more targeted training.

ACKNOWLEDGMENTS

We thank Darrell Johnson, Dawson Dill, Hyrum Osmond, and Jared Bishop for creating Olaf’s animations, Heather Green and Sarah Skinner for costuming, Robert Michel for creating the control interface, and Christian Schumacher for helping with visuals.

REFERENCES

- [1] J. Hurst, "Walk this way: To be useful around people, robots need to learn how to move like we do," *IEEE Spectrum*, vol. 56, no. 3, pp. 30–51, 2019.
- [2] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [3] N. Rudin, J. He, J. Aurand, and M. Hutter, "Parkour in the wild: Learning a general and extensible agile locomotion policy using multi-expert distillation and rl fine-tuning," *arXiv preprint arXiv:2505.11164*, 2025.
- [4] T. Zhu, M. S. Ahn, and D. Hong, "Artemis: An open-source, full-sized humanoid robot for dynamic locomotion," in *2025 IEEE-RAS 24th Int. Conf. Humanoid Robots*, 2025, pp. 269–276.
- [5] M. Fujita, "Aibo: Toward the era of digital creatures," *Int. J. Robotics Research*, vol. 20, no. 10, pp. 781–794, 2001.
- [6] H. Liu, M. Zhu, A. M. F. Alvarez, Y. H. Lo, C. Ku, F. Parres, J. Quan, C. Togashi, A. Navghare, Q. Wang, and D. W. Hong, "From screen to stage: Kid cosmo, a life-like, torque-controlled humanoid for entertainment robotics," in *2025 IEEE-RAS 24th Int. Conf. Humanoid Robots*, 2025, pp. 677–684.
- [7] R. Grandia, E. Knoop, M. Hopkins, G. Wiedebach, J. Bishop, S. Pickles, D. Müller, and M. Bächer, "Design and control of a bipedal robotic character," in *Robotics: Science and Systems XX*, Jul. 2024.
- [8] H. Kozima, M. P. Michalowski, and C. Nakagawa, "Keepon: A playful robot for research, therapy, and entertainment," *Int. J. social robotics*, vol. 1, no. 1, pp. 3–18, 2009.
- [9] H. Robinson, B. MacDonald, N. Kerse, and E. Broadbent, "The psychosocial effects of a companion robot: a randomized controlled trial," *J. American Medical Directors Association*, vol. 14, no. 9, pp. 661–667, 2013.
- [10] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: system overview and integration," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 3, 2002, pp. 2478–2483 vol.3.
- [11] J.-H. Oh, D. Hanson, W.-S. Kim, Y. Han, J.-Y. Kim, and I.-W. Park, "Design of android type humanoid robot albert hubo," in *2006 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2006.
- [12] D. Gouaillier, V. Hugel, P. Blazevice, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *2009 IEEE Int. Conf. robotics and automation*, 2009, pp. 769–774.
- [13] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath, "Berkeley humanoid: A research platform for learning-based control," in *2025 IEEE Int. Conf. Robotics and Automation*, 2025, pp. 2897–2904.
- [14] S. Lohmeier, T. Buschmann, and H. Ulbrich, "Humanoid robot lola," in *2009 IEEE Int. Conf. Robotics and Automation*, 2009, pp. 775–780.
- [15] S. Seok, A. Wang, M. Y. Chuah, D. J. Hyun, J. Lee, D. M. Otten, J. H. Lang, and S. Kim, "Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot," *Ieee/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2014.
- [16] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2016, pp. 38–44.
- [17] J. Reher, W.-L. Ma, and A. D. Ames, "Dynamic walking with compliance on a cassie bipedal robot," in *2019 18th European Control Conference*. IEEE, 2019, pp. 2589–2595.
- [18] A. SaLoutos, E. Stanger-Jones, Y. Ding, M. Chignoli, and S. Kim, "Design and development of the mit humanoid: A dynamic and robust research platform," in *2023 IEEE-RAS 22nd Int. Conf. Humanoid Robots*, 2023, pp. 1–8.
- [19] A. Badri-Spröwitz, A. Aghamaleki Sarvestani, M. Sitti, and M. A. Daley, "Birdbot achieves energy-efficient gait with minimal control using avian-inspired leg clutching," *Science Robotics*, vol. 7, no. 64, 2022.
- [20] Q. Shi, J. Gao, S. Wang, X. Quan, G. Jia, Q. Huang, and T. Fukuda, "Development of a small-sized quadruped robotic rat capable of multi-modal motions," *IEEE Trans. on Robotics*, vol. 38, no. 5, pp. 3027–3043, 2022.
- [21] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, "Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *Adaptive mobile robotics*. World Scientific, 2012, pp. 483–490.
- [22] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Trans. Robotics*, vol. 33, no. 3, pp. 509–522, 2017.
- [23] G. Maloisel, R. Grandia, C. Schumacher, E. Knoop, and M. Bächer, "A versatile quaternion-based constrained rigid body dynamics," *ACM Trans. Graph.*, vol. 44, no. 4, Jul. 2025.
- [24] G. Maloisel, C. Schumacher, E. Knoop, R. Grandia, and M. Bächer, "Optimal design of robotic character kinematics," *ACM Trans. Graphics*, vol. 42, no. 6, pp. 1–15, 2023.
- [25] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Trans. Graphics*, vol. 36, no. 4, pp. 41:1–41:13, 2017.
- [26] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eadu5872, 2019.
- [27] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE Int. Conf. Robotics and Automation*, 2021, pp. 2811–2817.
- [28] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. Van de Panne, "Feedback control for cassie with deep reinforcement learning," in *2018 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2018.
- [29] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, pp. 143:1–143:14, 2018.
- [30] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems*, 07 2020.
- [31] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer, "Vmp: Versatile motion priors for robustly tracking motion on physical characters," in *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation*, ser. SCA '24, 2024, p. 1–11.
- [32] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbabu, C. Pan, Z. Yi, G. Qu, K. Kitani, J. K. Hodgins, L. Fan, Y. Zhu, C. Liu, and G. Shi, "ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills," in *Proc. of Robotics: Science and Systems*, June 2025.
- [33] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, 2024.
- [34] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter, "Attention-based map encoding for learning generalized legged locomotion," *Science Robotics*, vol. 10, no. 105, p. eadv3604, 2025.
- [35] F. Bjelonic, F. Tischhauser, and M. Hutter, "Towards bridging the gap: Systematic sim-to-real transfer for diverse legged robots," *arXiv preprint arXiv:2509.06342*, 2025.
- [36] Y. Zhang, Y. Yao, S. Liu, Y. Niu, C. Lin, Y. Yang, W. Yu, T. Zhang, J. Tan, and D. Zhao, "Quietpaw: Learning quadrupedal locomotion with versatile noise preference alignment," *arXiv preprint arXiv:2503.05035*, 2025.
- [37] R. Watanabe, T. Miki, F. Shi, Y. Kadokawa, F. Bjelonic, K. Kawaharazuka, A. Cramariuc, and M. Hutter, "Learning quiet walking for a small home robot," *2025 IEEE Int. Conf. Robotics and Automation*, pp. 15 285–15 291, 2025.
- [38] M. A. Hopkins, G. Wiedebach, K. Cesare, J. Bishop, E. Knoop, and M. Bächer, "Interactive design of stylized walking gaits for robotic characters," *ACM Trans. Graph.*, vol. 43, no. 4, Jul. 2024.
- [39] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference*, 2019, pp. 3420–3431.
- [40] C. Schumacher, E. Knoop, and M. Bächer, "A versatile inverse kinematics formulation for retargeting motions onto robots with kinematic loops," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 2021.
- [41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
- [42] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *Int. J. Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.