Yuljeon: Bringing a University Mascot to Life on an Open-Source Bipedal Platform

Hajong Kim and Sangho Kim

***Abstract*—Yuljeon is the mascot of Sungkyunkwan University. This paper presents the design of a walking Yuljeon robot and a plan for its reinforcement learning controller. The mascot's large head, short legs and loose garment are proportions unlike those of the open small bipeds, such as PhoneBot, that a student team can build on, and they constrain the mechanism and the controller together. We adopt the lower body of an open-source small biped and add a new torso, two arms and a light head frame that gives a fabric head cover its shape. The heavier upper body pushes the hip-roll load past the original servos, so we replace those two actuators and check mass, balance and clearance on the CAD model. The small servos that carry the head work under a textile costume, which is expected to trap their heat and to add contact forces on the moving body. To address these effects, we plan to extend the PhoneBot training setup with costume disturbances, mass randomization and a temperature constraint. Simulation and hardware evaluation of the controller will follow the build.**

***Index Terms*—Humanoid and bipedal locomotion; entertainment robotics; reinforcement learning**

I. Introduction

RESEARCH on legged robots has mostly measured progress in speed, agility and robustness on rough ground [1]–[3]. Robots made for entertainment are judged first by how they look and move [4]–[7], and a campus mascot robot faces the same test from visitors who know the character. A robot that walks stably with the wrong gait or silhouette has missed its purpose, so for such a robot the given appearance is a requirement that the engineering has to meet.

Fig. 1 shows the subject of this paper, Yuljeon, the mascot of Sungkyunkwan University. Its head fills about one third of its height, its torso and legs are short, and it wears a traditional garment and a cap. Each of these features has a mechanical cost: the head raises the center of mass (CoM) and loads the hip joints, and cloth over the legs brings drag, contact and trapped heat. The mechanism must also stay inside the soft outline of a plush figure, because visitors who know the mascot from dolls and drawings spot an exposed bracket or a wrong proportion immediately.

Our design keeps the walking hardware of an existing platform and builds the character around it. We adopt the complete lower body of PhoneBot [8], an open-source 13-DoF biped driven by low-cost DYNAMIXEL XL430 servos, because its legs are already designed and documented. On top of it we design a new torso, two single-axis arms, a removable head frame and a fused power system for the mascot's upper body. Fabric hangs only from fixed parts such as a pelvis-mounted hip guard, which keeps the costume off the moving legs. The head cover with the face, the cap and the costume are sewn separately and hung on the printed skeleton.

The authors are with SKKAI, the artificial intelligence society of Sungkyunkwan University, Republic of Korea.

Draft. Yellow TBD boxes and blue text mark values and statements that await the hardware build and experiments.



Fig. 1. **Yuljeon Concept.** Generated concept image of the intended appearance on campus, not a photograph of the robot. The robot is not yet built, and the current costume target is a knee-length vest instead of the long robe shown here.

For control, we start from the reinforcement learning (RL) pipeline of the PhoneBot walking policy and adapt it to the new body. The taller and heavier upper body changes the dynamics, so we will retrain the policies on a simulation model whose link properties come from our CAD model. A static estimate puts the hip-roll load above the simulated torque limit of the original servo, and we therefore replace those two servos and will model the new actuator separately in simulation. The costume touches the body in ways that are hard to simulate, and following Olaf [7] we will treat its effect as randomized disturbance forces. Since the garment is expected to trap heat, we will also adopt the actuator temperature model and barrier-style temperature reward of Olaf [7]. Conventional control handles the shoulders, the voice hardware and the operator functions, which the learned policy does not see.

The contributions of this paper are:

- **Mascot-Scale Skeleton:** a 620 mm, 15-DoF skeleton design that places a mascot's large head, torso and arms on the unchanged legs of an open-source biped, with replaced hip-roll actuators, a pelvis-fixed hip guard for the garment and a two-stage fused power system.
- **Computed Design Checks:** a set of checks on the CAD model for mass, CoM, hip-roll load, tip angles, yaw

clearance and fuse budget, with the main assumed values stated and the rules run as automated tests.

- **Costume-Aware Training Plan:** an adaptation of the PhoneBot training setup to a heavier upper body and a changed hip-roll actuator, which adds the costume disturbances and the actuator temperature constraint of Olaf [7]; its evaluation is still to come.

II. Related Work

Legged robots usually borrow a human or animal body plan, and their task fixes their proportions. Learned controllers have produced agile quadruped motion [1] and real-world humanoid walking [2], [3]. Recent humanoids are built for fall robustness under learning-based control [9], energy-efficient walking through passive dynamics [10], or proprioceptive actuation at miniature scale [11]. Yuljeon reverses this order: the mascot's drawn appearance comes first, and the walking hardware must fit inside that outline.

Entertainment robots have long been designed around a character, from the pet-like AIBO [4] to entertainment humanoids [5]. Recent work controls a bipedal robotic character that performs artistic motion [6] and builds a torque-controlled humanoid whose visual form is a design priority [12]. Olaf is brought to life with two asymmetric legs hidden under a soft foam skirt, animation-guided RL policies, and rewards that limit actuator temperature and footstep noise [7]. Printed soft skins can also make small robots safer to touch [13]. In Yuljeon, the character is mounted on an already published open platform with *low-cost* servos, so the legs stay fixed and every character change is made above the hip.

Open humanoids have made bipedal research accessible to small teams, from DARwIn-OP [14] and the printed Poppy [15] to ToddlerBot [16] and Berkeley Humanoid Lite [17]. Open Duck Mini rebuilds a small version of a character robot at low cost [18]. PhoneBot [8] is a 1.8 kg, 13-DoF biped that uses a smartphone for sensing and computing, costs about USD 400 without the phone, and releases its hardware design. Its authors name the missing arms and the limited servo torque as limitations. We build on its legs and add what a mascot needs: arms, a head, a costume and the mass that comes with them.

Massively parallel simulation has made RL practical for legged locomotion [19], supported by GPU frameworks [20]–[22] and the MuJoCo engine [23], and policies are commonly optimized with PPO [24]. Imitation of reference motion gives simulated characters and robots a desired style [25]–[27], and artist tools produce stylized gaits within a robot's limits [28]. Randomized rendering [29] and dynamics [30] help models trained in simulation transfer to the real world, as do learned actuator models [1]. Control barrier functions encode safety limits [31], and Olaf turns this idea into a temperature reward for its small, costume-covered neck actuators [7]. We start from the PhoneBot pipeline and add the temperature constraint and costume disturbances of Olaf.

III. Workflow

The design started from two constraints: the robot must match the mascot's real proportions, and its legs must come from a platform that already walks. The published PhoneBot assembly supplied the legs, pelvis, torso-yaw bearing and battery seat, which we left as released apart from the hip-roll servos and a pilot hole in one pelvis part (blue in Fig. 2). The torso, arms, head frame, hip guard and power system are new and written as parametric CAD code (vermillion). After each change, scripts recompute mass, CoM, clearances and electrical budgets from the CAD model, and 543 design-rule tests cover these checks. We planned the textile exterior from the CAD dimensions of the skeleton and deferred its sewing-level design to a later phase. Sec. IV details the hardware.

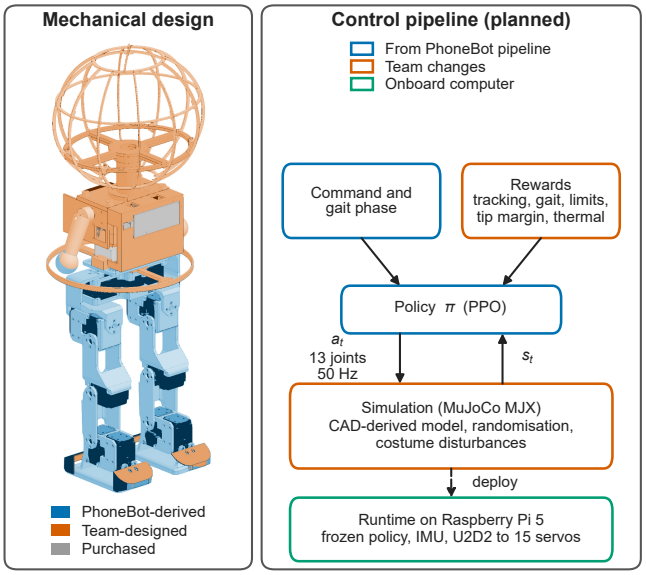


Fig. 2. **Design and Control Overview.** Left: CAD render of the skeleton, with the lower body adopted from PhoneBot [8] in blue and the torso, arms and head frame designed by the team in vermillion. Right: the planned control pipeline, in which the command and gait phase, the reward terms and the costume disturbances drive policy training in simulation. The trained policy is to run on the onboard computer and send joint targets to the servos.

The 13 leg and torso-yaw joints will be driven by a learned policy (Sec. V), and conventional control runs the shoulders, voice hardware and operator functions (Sec. VI). For simulation, the CAD model already exports link masses, inertias and CoM positions. These properties will replace the corresponding values in the PhoneBot simulation model, which will also gain randomized costume forces and an actuator temperature state. The reward will combine the PhoneBot tracking and gait terms with penalties for joint range, actuator temperature and the reduced tip margin, and which motion modes receive separate policies is TBD: policy set.

At runtime, an operator drives the robot with a gamepad and holds a tethered emergency stop, while the onboard computer runs the policy and the interaction software (Sec. VII-C).

IV. Mechatronic Design

Yuljeon is 620 mm tall from the sole to the top of the fabric-covered head, without the cap. The CAD model of the Raspberry Pi 5 build totals 2,440.2 g, including estimated cap, head cover and wiring and a 91.8 g costume estimated for the earlier ankle-length robe, not the current vest and trousers. With six joints per leg (hip yaw, hip roll, hip pitch, knee,

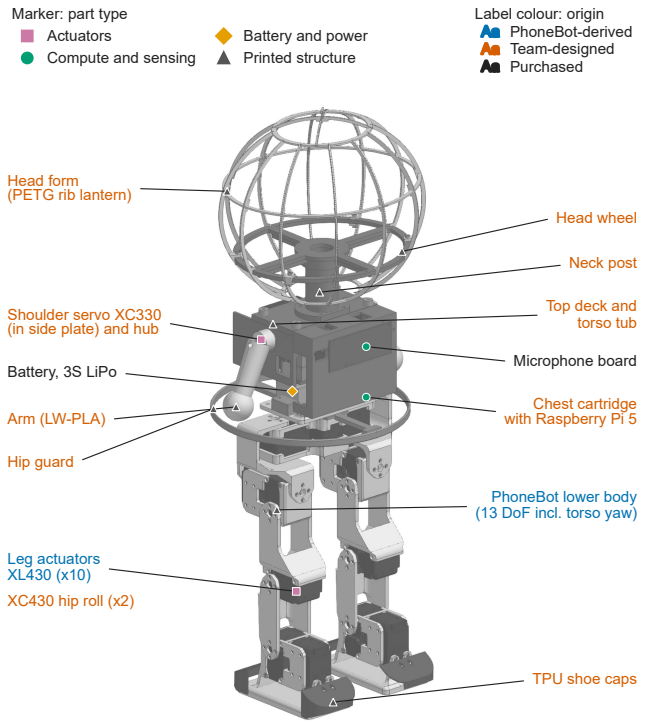


Fig. 3. **Mechatronic Layout.** Annotated CAD view of the designed skeleton and internals with the textile exterior removed. Marker shapes give the part type (actuators, compute and sensing, power, printed structure), and label colors give the origin: blue for PhoneBot-derived, vermillion for team-designed and black for purchased parts.

ankle pitch and ankle roll), the torso yaw and one shoulder pitch per arm, the robot has 15 DoF (Fig. 3). Eleven joints use DYNAMIXEL XL430-W250-T servos, the two hip-roll joints use XC430-W240-T and the shoulders use XC330-T288-T. A Raspberry Pi 5 in a chest cartridge runs the software, and a Jetson Orin Nano Super uses the same cartridge mount, in a deeper box, for a later phase. Compared with PhoneBot at 1.8 kg and 0.48 m [8], the same legs now carry a taller and heavier upper body.

Lower Body from PhoneBot: The feet, leg links, hip brackets, pelvis, servo positions, torso-yaw thrust bearing and battery seat are kept as published, except one PelvisTop, reprinted with a 2.1 mm pilot hole for the rear lock screw of the hip guard. The phone, phone holder and electronics frame are removed. At hip roll, XC430 servos replace the XL430 in the same case, horn and screw pattern. Their side holes are only 3.0 mm deep, so the case screws are shortened to M2.5×4, and each servo adds 7.8 g. TPU caps extend each foot by 15 mm at the toe and 10 mm at the heel. Their undersides start 3 mm above the ground and rise at 20°, so they touch the ground only when the foot tilts beyond that angle, while the sole plate and the 45 mm foot gap stay unchanged.

Torso and Arms: The torso, a U-shaped PLA tub 114 mm wide with 4 mm walls and open at the front and top, houses the power hub, relay and fuses. A 6 mm lid carries the head and is about three times stiffer in bending than a 4 mm plate. The computer sits in a removable PETG cartridge, held by forked feet on three existing screws and two captive M3 lock bolts. To swap the computer, only this box changes: it is 26.5 mm deep for the Pi 5 and 40 mm for the Jetson. Each arm is a 68 mm LW-PLA part with a 25° splay, keyed to the hub of an XC330 that sits inside the side plate and held by one screw. At 6.0 g, the arm is meant to break first in a hard hit.

Head Frame: The fabric-covered head is an ellipsoid of 232×206×208 mm, about one third of the robot's height. A printed PETG "lantern" of 12 meridian ribs, a belt hoop, crown and chin rings and two crown arcs gives the cover its shape. The frame lies 5 mm inside the envelope on every side (222×196×198 mm) to leave room for foam and fabric, prints flat without support and has a computed mass of 35.1 g. It mounts on a spoked head wheel on a tubular neck post, and three screws release the whole head without opening the torso. The head center sits 18 mm forward of the yaw axis to pull the CoM forward, and the computed mass above the torso-yaw joint is 1,123.7 g.

Power and Protection: A 3S LiPo pack of 2200 mAh and 137 g sits in the reused PhoneBot battery seat and is used between 9.6 and 12.6 V. A 20 A main fuse follows the battery connector, after which the circuit splits into three branches. In the motor branch, a 10 A fuse and a relay lead to the servo hub. Resettable fuses feed both ends of each leg chain, because the estimated support-leg current of 2.47–3.09 A reaches the 3 A connector rating. Separate logic and control branches feed the computer and the relay coil, so a motor fault leaves the computer running. Known limits remain: there is no hardware cut-off against over-discharge, the relay has a single contact, and a full charge of 12.6 V exceeds the 12.0 V maximum rating of the XL430.

Textile Exterior: The exterior is fabric: a head cover with the face and a foam liner, a cap and a costume. The current target is a knee-length blue vest, white trousers and black shoe covers. Fabric hangs only on parts that do not move with the leg links, since cloth on the links would add forces missing from the simulation. A pelvis-fixed hoop with a radius of about 100 mm and a mass of 18.0 g carries the trousers, and its inner face stays 4.4 mm outside the circle swept by the hip-yaw bracket. Cloth that turns with the torso thus slides over a fixed rim. Vents in the cartridge floor, lid and head frame form an air path for a short garment, whose effect under the costume is unchecked. For service, the battery slides out through a side slit under the raised right arm, and the cartridge comes off after two bolts.

Computed Checks: For the scripted checks in Tab. I, printed masses assume 80% of solid density, the PhoneBot lower-body masses are fitted to its published 1.8 kg, and currents assume load factors of 0.90 at hip roll and 0.20 elsewhere, a buck efficiency of 0.85 and a 0.5 A USB load. The hip-roll index is the total weight times the 52.5 mm half hip spacing, a rough static torque for single support (Fig. 4). For the original PhoneBot this index is 0.927 N m, equal to the 0.93 N m torque limit of its simulation model, and Yuljeon reaches 1.257 N m, which motivated the servo change. The total mass is about 36% above PhoneBot and 363.6 g above the walking target of our design review, and the listed reduction candidates do not close this gap. Simulation training will decide whether the 2.4 kg robot walks. With the torso yawed by ±90°, the backward tip angle drops to 6.7 and 7.0° for the Pi 5 build

TABLE I
DESIGN CHECKS COMPUTED FROM THE CAD MODEL. VALUES FOR THE RASPBERRY PI 5 AND JETSON BUILDS. MASSES INCLUDE ESTIMATED COSTUME, CAP, HEAD COVER AND WIRING, AND THE PESSIMISTIC CASE ASSUMES A 150 G HEAD COVER, A 185.6 G GARMENT AND 100 G OF WIRING. NO VALUE IS MEASURED.

Check	Pi 5	Jetson
Total mass (g)	2,440.2	2,524.9
Pessimistic mass (g)	2,659.0	2,743.7
CoM height (mm)	260.1	263.6
CoM behind ankle-pitch axis (mm)	5.1	3.2
Hip-roll index (N m)	1.257	1.300
vs. XL430 sim limit 0.93 N m	1.35×	1.40×
vs. XC430 stall 1.7 N m	0.74	0.76
Tip angle back / fwd / side (°)	8.3/15.5/17.5	8.6/14.9/17.3
Arm-tip yaw clearance, req. 5 mm (mm)	6.5	6.5
Hip-guard mount gap, waiver 3.0 mm (mm)	3.10	3.10
Part interference >1 mm³	0	0
Main fuse peak share	0.83	0.96
Yaw-bearing overturn accel. (m/s²)	0.67	0.16
Continuous walking time (min)	14–17	11.2–15.4

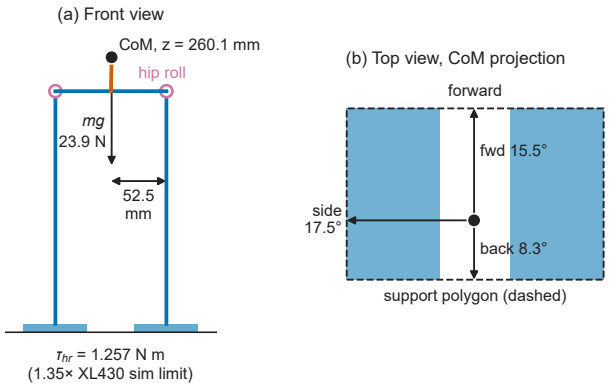


Fig. 4. **Hip-Roll Load.** (a) Front-view schematic of the CoM at a computed height of 260.1 mm and the 52.5 mm half hip spacing that set the static hip-roll index of the Pi 5 build. (b) Top view of the CoM projection in the support polygon, with the computed backward, forward and side tip angles of the Pi 5 build.

(6.6 and 6.8° for the Jetson), below the 8.0° requirement of our test plan. The lever-arm loads of the other eleven XL430 joints are not yet computed, and interference checks exclude pairs within the original PhoneBot geometry.

V. REINFORCEMENT LEARNING

No policy has been trained for Yuljeon yet, and this section gives the planned formulation. Each motion mode that Yuljeon needs will get its own policy, starting with walking; whether the PhoneBot standup and heading policies are retrained as well is TBD: mode set. We model every mode as a Markov decision process: given an observation $\mathbf{o}_t$ and a command $\mathbf{c}_t$, the policy π outputs an action $\mathbf{a}_t$ and receives a reward r_t.

PhoneBot expresses its observations in the frame of the phone IMU [8]. Yuljeon has no phone, and its IMU sits on the inside of the right side plate, so we define the proprioceptive state in the IMU frame $\mathcal{I}$ as

$$\mathbf{s}_t = \left({}^{\mathcal{I}}\boldsymbol{\omega}_t,\ {}^{\mathcal{I}}\mathbf{g}_t,\ \mathbf{q}_t - \mathbf{q}_{\text{home}},\ \dot{\mathbf{q}}_t\right), \tag{1}$$

where ${}^{\mathcal{I}}\boldsymbol{\omega}_t$ is the angular velocity, ${}^{\mathcal{I}}\mathbf{g}_t$ the gravity direction, $\mathbf{q}_t \in \mathbb{R}^{13}$ the positions of the leg and torso-yaw joints, $\mathbf{q}_{\text{home}}$ the home pose and $\dot{\mathbf{q}}_t$ the joint velocities.

The IMU sits at $(-48.6, 0, 359)$ mm, off the sagittal plane. PhoneBot augments each PPO update with mirrored samples, which assumes a left-right symmetric robot [8]. The computed CoM lies 0.7 mm from the midline, but the sign rules for the moved IMU and new hip-roll mounting are TBD: mirror check. We will use the IMU's game rotation vector, which ignores nearby magnets, and confirm its axis signs on hardware.

A. Command and Gait Reference

PhoneBot walks without a motion reference, and its policy tracks a velocity command with a periodic gait phase [8]. We keep this structure, with command and phase

$$\begin{aligned} \mathbf{c}_t &= (v_x, v_y, \omega_z), \quad v_x, v_y \sim \mathcal{U}[-v_{\max}, v_{\max}], \\ &\qquad\qquad \omega_z \sim \mathcal{U}[-\omega_{\max}, \omega_{\max}], \\ \boldsymbol{\phi}_{t+1} &= \boldsymbol{\phi}_t + 2\pi f \Delta t \pmod{2\pi}, \\ \boldsymbol{\Phi}_t &= \begin{cases} (\cos\boldsymbol{\phi}_t, \sin\boldsymbol{\phi}_t), & \|\mathbf{c}_t\| > \epsilon, \\ (\cos\boldsymbol{\phi}_0, \sin\boldsymbol{\phi}_0), & \text{otherwise}, \end{cases} \end{aligned} \tag{2}$$

where $v_{\max}$ and $\omega_{\max}$ bound the sampled command, $\boldsymbol{\phi}_t = (\phi_L, \phi_R)$ holds the phases of the two feet, f is the gait frequency and $\Delta t = 0.02$ s is the control period. Cosine and sine act elementwise, and for a stand command, $\|\mathbf{c}_t\| \le \epsilon = 0.05$, the phase input is computed from $\boldsymbol{\phi}_0 = (\pi, \pi)$. PhoneBot samples $v_{\max} = 0.5$ m/s and $\omega_{\max} = 1.0$ rad/s, draws a zero command with probability 0.1 and evaluates its gait at $f = 1.5$ Hz. For the heavier Yuljeon, these values are TBD: command range, f. A stylized walking reference from an artist gait tool [28], tracked with an imitation reward [25], could bring the gait closer to the character, and whether to add one is TBD: style reference.

B. Policy

The action $\mathbf{a}_t \in \mathbb{R}^{13}$ is a joint-position residual about the home pose, applied at 50 Hz as in PhoneBot. The target $\hat{\mathbf{q}}_t = \text{clip}(\mathbf{q}_{\text{home}} + \mathbf{a}_t + \boldsymbol{\delta})$, with a per-episode offset $\boldsymbol{\delta}$, is turned into torque by the servo model of Sec. V-E. The actor observes

$$\mathbf{o}_t = \left(\tilde{\boldsymbol{\omega}}_t,\ \tilde{\mathbf{g}}_t,\ \mathbf{c}_t,\ \tilde{\mathbf{q}}_t - \mathbf{q}_{\text{home}},\ \dot{\tilde{\mathbf{q}}}_t,\ \mathbf{a}_{t-1},\ \boldsymbol{\Phi}_t,\ \tilde{\mathbf{T}}_t\right), \tag{3}$$

where the tilde marks noisy signals, $\tilde{\mathbf{q}}_t$ includes the simulated backlash of the observed joint positions, and $\tilde{\mathbf{T}}_t$ holds the temperatures of the modeled actuators (Sec. V-D). Without $\tilde{\mathbf{T}}_t$, this is the 52-dimensional PhoneBot actor input, and the critic receives privileged signals as in PhoneBot [8]. The shoulder joints are in the simulation model but not in the action, and will follow random motions within their limits in training.

C. Reward Formulation

As for Olaf [7], the reward is a sum of groups, here *tracking*, *regularization*, *gait*, *pose* and *limits*:

$$\begin{aligned} r_t &= \Delta t \left(r^{\text{track}} + r^{\text{reg}} + r^{\text{gait}} + r^{\text{pose}} + r^{\text{lim}}\right), \\ r^g &= \textstyle\sum_{i \in g} w_i r_i. \end{aligned} \tag{4}$$

Tracking pays for following the velocity command and staying upright, regularization keeps body rates, torque and action

TABLE II
REWARD TERMS. PENALTIES ARE WRITTEN AS POSITIVE QUANTITIES AND RECEIVE NEGATIVE WEIGHTS IN TAB. IV, AS IN PHONEBOT [8]. TERMS THAT PHONEBOT NAMES WITHOUT PRINTING A FORMULA ARE DESCRIBED IN WORDS. $u = |\tau|/\tau_{\text{lim}}$, $\sigma = 0.1$, AND h^* IS THE SWING-HEIGHT TARGET. THE TEMPERATURE AND BARRIER JOINT-LIMIT TERMS FOLLOW OLAF [7].

Name	Term
	Tracking
Linear velocity	$\exp(-\|\mathbf{v}_{xy} - \mathbf{c}_{xy}\|^2/\sigma)$
Yaw velocity	$\exp(-(\omega_z - c_z)^2/\sigma)$
Alive	1
	Regularization
Vertical velocity	v_z^2
Roll/pitch rate	body roll and pitch rates
Orientation	$\|\mathbf{g}_{xy}\|^2$
Torque ℓ_1, ℓ_2	$\|\boldsymbol{\tau}\|_1$, mean of τ^2
Torque utilization	mean of $\text{clip}((u-0.8)/0.2, 0, 1)^2$
Action rate	change of action between steps
Joint velocity, acc.	joint velocity, joint acceleration
	Gait
Feet air time	$(t_{\text{air}} - 0.2)_+$ at first contact while moving
Feet phase, corner phase	contact timing against $\boldsymbol{\Phi}_t$
Feet height, clearance	swing height against h^*
Feet corner clearance	clearance of the sole corners
Feet parallel, slip	foot alignment, sliding in contact
Feet distance	$[d_{\min} - d_{\text{feet}}]_+$, $d_{\min}$ TBD
Foot collision	$\mathbb{1}[\text{contact}(L, R)]$
	Pose
Pose to home	weighted deviation from $\mathbf{q}_{\text{home}}$
Hip roll, knee deviation	deviation from the home angle
Stand still	joint motion while $\|\mathbf{c}\| \le \epsilon$
	Limits
Joint limits	PhoneBot term or barrier (8), TBD: choice
Temperature	$\|\min(-\dot{\mathbf{T}} + \gamma_T(T_{\max} - \mathbf{T}), \mathbf{0})\|_1$
Tip margin	TBD: form

changes small, and the gait and pose terms of PhoneBot [8] set foot timing and posture. The limits group holds the constraints that matter for our design: joint range, actuator temperature and the backward tip margin that shrinks when the torso yaws. The terms appear in Tab. II, and Tab. IV gives the weights, each open Yuljeon value beside its PhoneBot value.

Three gait terms need new settings. The shoe caps make each foot 25 mm longer, so we revisit the feet-distance threshold of 0.05 m used by PhoneBot, keep the foot-corner terms on the original sole plate and add the caps to the collision set. Poses in which a shin rises above horizontal and meets the hip guard are handled by TBD: block or penalty.

The tip-margin term addresses a computed weakness: with the torso yawed, the backward tip angle falls to 6.6 to 7.0° (Sec. IV). This region is either penalized or excluded from the command range, TBD: reward or command limit. We will also log the horizontal acceleration of the upper body against the computed yaw-bearing limit of 0.67 m/s^2. Whether to add an impact term like that of Olaf, which reduced walking noise [7], is TBD: impact term (Sec. VIII-C).

D. Thermal Modeling

Under the costume, the electrical power drawn while walking, estimated at 69 to 86 W, ends up as heat inside the garment. The servo limits are 72 °C for the XL430, 80 °C for the XC430 and 70 °C for the XC330, and the temperature rise inside the costume has not been measured. We write the limit $T \le T_{\max}$ of each modeled actuator as a barrier condition [31], as Olaf does for its neck actuators [7]:

$$h_T = T_{\max} - T \ge 0, \tag{5a}$$

$$\dot{h}_T + \gamma_T h_T \ge 0, \tag{5b}$$

$$-\dot{T} + \gamma_T (T_{\max} - T) \ge 0. \tag{5c}$$

Far from the limit, temperature may rise freely; as T approaches $T_{\max}$ the condition drives its rate of rise toward zero.

The violation of (5c) enters the limits group as a penalty (Tab. II). Training needs a simulated temperature, so we adopt the first-order model of Olaf [7], in which Joule heating from torque drives the rise:

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

where α, β and T_{amb} are fitted for each actuator type. Heat from the electronics inside the garment reaches the model only through T_{amb}, and whether T_{amb} should depend on the costume state is TBD: ambient model. Olaf models only its neck actuators, which Yuljeon does not have, so the modeled joints are TBD: joint set; the hip-roll joints, with the highest computed static load we have analyzed, are the first candidates. Sec. VII-A describes the planned fit.

E. Joint and Torque Limits

Joint limits can be written with the same barrier pattern, as Olaf does [7]. With a margin q_m inside the physical limits $[q_{\text{lo}}, q_{\text{hi}}]$,

$$h_q = \begin{cases} q - (q_{\text{lo}} + q_m), & \text{lower}, \\ (q_{\text{hi}} - q_m) - q, & \text{upper}, \end{cases} \tag{7}$$

$$\dot{h}_q + \gamma_q h_q \ge 0 \quad \text{for each joint and bound}, \tag{8}$$

where q_m and γ_q are TBD: q_m, γ_q; Olaf uses 0.1 rad and 20 [7]. Torque follows the PhoneBot servo envelope with a 0.93 N m peak below 2.0 rad/s [8]. We keep this 0.93 N m limit on all joints, including the XC430 hip-roll joints, until the XC430 torque-speed curve is measured on the bench between 9.6 and 12.6 V. Voltage-dependent torque curves for both servo types will follow that measurement.

VI. SHOW FUNCTIONS

Three subsystems run without the learned controller: the shoulders, the voice hardware and the operator functions. At 6.0 g, an arm barely changes the body dynamics, and the policy will meet its motion only as a disturbance (Sec. V-B). A shoulder turns about one fore-aft axis, and the arm angle equals the servo angle, so the internal position loop of the XC330 will drive it directly. The limits of 135° forward and 150° back come from a sweep check, which finds the hand touching the head envelope from 160° forward and from 170° back. The shoulder servos share the battery-voltage bus with the leg servos.

For interaction, a two-microphone USB board sits in front of the chest cartridge and also drives a 4 Ω, 5 W speaker on the rear plate. Dialogue with students is planned for the university festival in May 2027, and speech-to-action control for autumn

TABLE III
THERMAL MODEL PARAMETERS. OLAF VALUES ARE FOR ITS NECK ACTUATORS AND ARE GIVEN FOR REFERENCE; YULJEON VALUES AWAIT THE DRESSED MEASUREMENT.

Parameter	Olaf neck [7]	Yuljeon
Model (6)		
α	0.038	TBD
β	0.377	TBD
T_{amb}	43.94	TBD
Reward (5c)		
$T_{\max}$ (°C)	80	TBD
$[T_{c,\min}, T_{c,\max}]$ (°C)	[70, 85]	TBD
γ_T	0.312	TBD
Modeled joints	neck	TBD

2027. Gestures that combine the arms with speech are planned but not yet designed. Voice announcements on low battery and on a forced stop are defined as operating rules, and their software is not yet written.

An operator will drive the robot with a Bluetooth gamepad that sends velocity commands and triggers content. A tethered emergency stop of about 3 m carries two normally closed buttons in series in the coil circuit of the motor relay. Pressing either button removes power from all 15 servos while the computer stays on, so the robot collapses and has to be held by a stand or a safety line. Before torque is enabled, the operating procedure requires the operator to press the button, confirm that the servo bus is dead, release it and confirm that power returns. The costume sleeves may load the arms, and the sleeve attachment and any compensation for it are TBD: sleeve design.

VII. IMPLEMENTATION DETAILS

A. Thermal Model and Reward

We will fit (6) with the procedure of Olaf [7]: during a 10 min run in the costume we log servo and computer temperatures, discretize the model with an explicit Euler step at the policy rate and obtain α, β and T_{amb} by least squares. The fitted values will go into Tab. III, and Sec. VIII-B describes how they will be tested. After the measurement, $T_{\max}$ will be fixed below the servo ratings of 72 and 80 °C, at TBD: $T_{\max}$.

Observed and rewarded temperatures are clipped to $[T_{c,\min}, T_{c,\max}]$ = TBD: clip range, which keeps the range seen in training small. With zero torque at the upper clip bound, (6) and (5c) remain consistent when $\gamma_T \leq \alpha(T_{c,\max} - T_{\mathrm{amb}})/(T_{c,\max} - T_{\max})$, and we will choose γ_T within this bound, as was done for Olaf [7].

B. Training

Policies will be trained with PPO [24] in an asymmetric actor-critic setup, with observation noise, pushes, randomized actuator parameters [30] and the costume disturbance forces, whose ranges are TBD: force ranges. Mass randomization will cover a head cover of 85 to 150 g, a garment of 43 to 186 g, wiring of 40 to 100 g and shifts of the CoM, at total masses of 2.4 to 2.7 kg. Network sizes are TBD: network, and Tab. IV lists the reward weights.

Training will run in MuJoCo with MJX [22], [23] on TBD: GPU with TBD: count parallel environments for TBD: iterations, taking TBD: time. If no policy walks at this mass, we will decide on mass reductions before the upper body is built.

TABLE IV
REWARD WEIGHTS. PHONEBOT WALKING WEIGHTS [8] ARE THE STARTING VALUES. YULJEON WEIGHTS STAY OPEN UNTIL TRAINING RUNS, AND THE LAST TWO TERMS ARE NEW.

Name	PhoneBot	Yuljeon
Linear velocity	4.0	TBD
Yaw velocity	4.0	TBD
Alive	0.5	TBD
Vertical velocity	−0.2	TBD
Roll/pitch rate	−0.1	TBD
Orientation	−10.0	TBD
Torque ℓ_1 / ℓ_2	−0.1 / −1.0	TBD
Torque utilization	−2.0	TBD
Action rate	−0.2	TBD
Joint acc. / vel.	$-2 \cdot 10^{-6}$ / −0.002	TBD
Joint limits	−1.0	TBD
Pose to home	−1.5	TBD
Hip roll / knee deviation	−1.0 / −0.005	TBD
Stand still	−0.1	TBD
Feet air time	10.0	TBD
Feet phase / corner phase	1.2 / 1.5	TBD
Feet clearance / height	−5.0 / −5.0	TBD
Feet corner clearance	−10.0	TBD
Feet parallel / slip	−5.0 / −10.0	TBD
Feet distance	−1.0	TBD
Foot collision	−1.0	TBD
Temperature	n/a	TBD
Tip margin	n/a	TBD

C. Runtime

At runtime, the frozen actor will run on the Raspberry Pi 5 in the chest cartridge at 50 Hz and send joint targets through a USB-to-TTL interface to the servo bus. The BNO085 IMU connects to the computer over SPI. The state estimator, any upsampling of the joint targets and an output filter are TBD: runtime chain. The Jetson alternative is capped at 25 W in its power settings. For the Pi 5 build, the estimated battery current while walking is 6.25 to 7.71 A, which gives 14 to 17 min of continuous walking per charge if 80% of the capacity is usable.

Operation will follow three stages:

- *Start-up:* the interlock of Sec. VI completes before servo torque is enabled, and with torque off the robot draws an estimated 1.31 A from the battery.
- *Operation:* gamepad commands select the policy and set the velocity command, and voice content plays through the speaker.
- *Protective stop:* the operating rules set a warning at 10.5 V and a forced stop at 9.9 V so that the pack never drops below 9.6 V, and the tethered emergency stop acts in hardware at any time.

VIII. RESULTS

Once the robot is built, we will evaluate the hardware design qualitatively: whether the garment stays clear of the moving legs during walking and torso yaw, whether a hard hit breaks the inexpensive arm first, and whether the battery, cartridge and head can be serviced as designed. We will also compare the computed values of Tab. I with weighed masses and measured tip angles.

Secs. VIII-A to VIII-D cover the controller and its deployment.

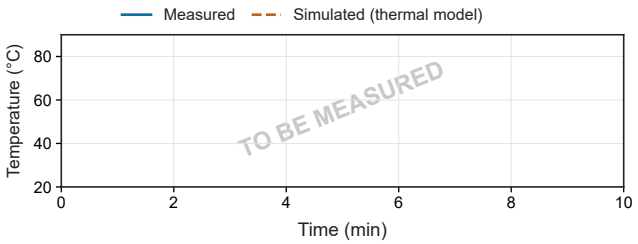


Fig. 5. **Held-Out Temperature Prediction.** Logged and predicted temperature of the TBD: joint actuator on a TBD: duration trajectory not used for fitting. No data exist yet.

A. Tracking Performance

Each policy will run for TBD: duration while its velocity command sweeps the whole range, with body velocity measured by TBD: method; the metric is the mean absolute gap between command and measurement. For walking, the planar and yaw gaps are TBD ± TBD m/s and TBD ± TBD rad/s; for standing, where the command is a fixed pose, the joint gap is TBD ± TBD °.

As a first step, a legs-only build will reproduce PhoneBot walking before the upper body is mounted, and its outcome is TBD. We will then compare policies trained with and without the costume disturbance forces on the dressed robot, TBD: result.

B. Thermal Model Evaluation

The fitted model will be tested on a TBD: duration trajectory not used for fitting, starting from the logged initial temperature, and Fig. 5 will lay the predicted curve over the logged one. The mean absolute difference between the two is TBD °C. We will also record the temperature rise inside the costume relative to the undressed robot, TBD °C after TBD: duration, a quantity that Olaf does not report.

A policy updates every 20 ms, but servo temperature drifts over minutes, so the reward for keeping cool arrives long after the actions that caused the heat. The temperature reward will be tested on the TBD: joint actuator, selected because TBD: reason, by running TBD: test motion with one policy trained with the reward and one without. This motion should load the chosen joint steadily enough that its temperature rises within one battery charge.

Fig. 6 will show these three traces for both policies. The run without the reward will be stopped at TBD °C, reached after TBD s. For the run with the reward, we will read the temperature it settles toward, TBD °C, and the tracking error the reward costs, TBD. The squared torque will show whether torque demand changes as the limit nears, TBD: observation. A separate TBD: duration run with the reward gives the final temperature, TBD °C, and the tracking error, TBD. Because the estimated continuous walking time per charge is 14 to 17 min, such a long run needs TBD: battery changes or tethered supply.

C. Foot Impact

Should the impact term of Sec. V-C be adopted, two hardware runs of TBD: duration each, one per policy, will be recorded with the microphone at TBD: position. We will report the change in mean noise level, TBD dB, and in velocity tracking error, TBD. Because the TPU shoe caps and the shoe covers may also change the impact sound, we will measure their effect in a separate run, TBD.

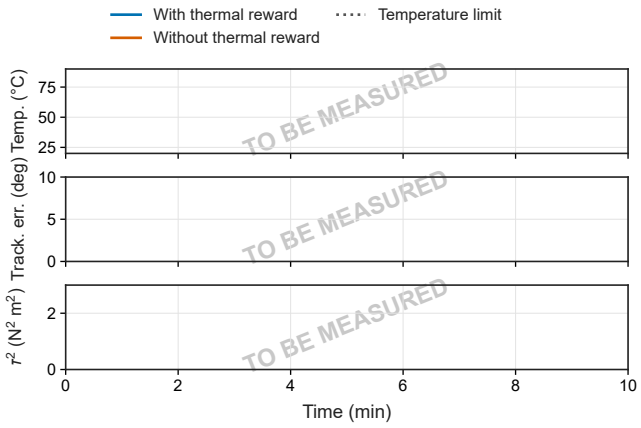


Fig. 6. **Effect of the Temperature Reward.** Actuator temperature (top), joint tracking error (middle) and squared torque (bottom) for a policy that sees the temperature reward and one that does not. No data exist yet.

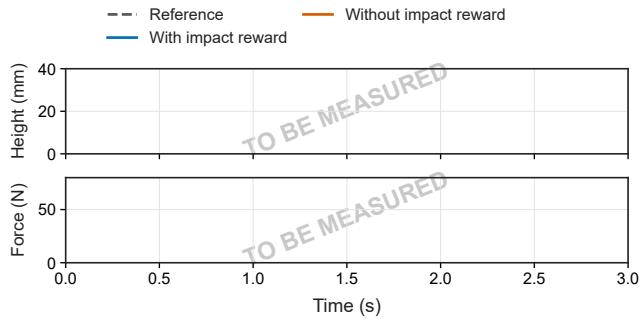


Fig. 7. **Swing Height and Landing.** Commanded and reached foot height (top) and vertical foot contact force (bottom), for a policy with an impact term and one without. No data exist yet.

Fig. 7 will overlay the commanded swing height, the foot height reached by each policy and the vertical contact force over several steps. From it we will read the landing velocity that the term removes, TBD m/s, and any smoothing of the swing.

D. System Deployment

The robot is scheduled to appear at the May 2027 festival, in autumn 2027 and at the February 2028 entrance ceremony. Operating hours, the number of events, and whether any servo reaches its temperature limit during a performance are TBD. During each event, we will log battery voltage, chain currents and servo temperatures and compare them with the estimates of Sec. IV.

IX. Conclusion

Yuljeon places a university mascot's large head, arms and textile costume on the legs of an open-source biped whose servos stall at 1.4 N m. So far the design exists only as CAD code and computed checks, and no part of this paper has yet been built, trained or measured.

The PhoneBot lower body stays as released except for hip-roll servos chosen from a static load estimate. The new parts, a removable head frame, a pelvis-fixed hip guard and a two-stage fused power system, were checked by computation on the CAD model. The control plan adapts the PhoneBot training setup with a split actuator model, mass randomization, costume disturbances and a barrier-based temperature reward for actuators that work under fabric.

The computed checks found the hip-roll overload and the smaller tip margin under torso yaw before any part was bought, while both could still be fixed by editing the CAD code. Such checks pay off most on a platform with little torque to spare, and we expect the approach of fitting a mascot onto an existing open biped and checking it in CAD first to carry over to other mascot robots.

The first open question is whether the 2.4 kg robot walks at all, which the planned training must answer. Garment contact forces and trapped heat, which the plan reduces to random disturbances and an ambient temperature, call for a cloth model that the simulation does not yet have. A bench-measured torque curve of the hip-roll servo would let the simulation drop the conservative limit it now applies.

Acknowledgment

Funding: TBD: funding source. The lower-body geometry is the PhoneBot design by Hou *et al.* [8], used under the CC BY-NC 4.0 license stated at https://phonebot.dev. We used an AI assistant (Claude, Anthropic) for drafting and editing this manuscript, and the authors reviewed all content.

References

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

[2] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[3] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 840–888, 2025.

[4] M. Fujita, "AIBO: Toward the era of digital creatures," *The International Journal of Robotics Research*, vol. 20, no. 10, pp. 781–794, 2001.

[5] S. Silverstein and K. Yamane, "Humanoid robots for entertainment," in *Humanoid Robotics: A Reference*, A. Goswami and P. Vadakkepat, Eds. Dordrecht: Springer Netherlands, 2019, pp. 2599–2615.

[6] R. Grandia, E. Knoop, M. A. 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*, 2024.

[7] D. Müller, E. Knoop, D. Mylonopoulos, A. Serifi, M. A. Hopkins, R. Grandia, and M. Bächer, "Olaf: Bringing an animated character to life in the physical world," *arXiv preprint arXiv:2512.16705*, 2025.

[8] R. Hou, Q. Wang, D. Koh, and D. W. Hong, "PhoneBot: A low-cost open humanoid robot platform reusing smartphones," *arXiv preprint arXiv:2610.08737*, 2026.

[9] 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 International Conference on Robotics and Automation (ICRA)*, 2025, pp. 2897–2904.

[10] B. Xia, B. Li, J. Lee, M. Scutari, and B. Chen, "The Duke Humanoid: Design and control for energy-efficient bipedal locomotion using passive dynamics," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 6579–6586.

[11] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong, "Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8547–8553.

[12] 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 International Conference on Humanoid Robots (Humanoids)*, 2025, pp. 677–684.

[13] J. Kim, A. Alspach, and K. Yamane, "3D printed soft skin for safe human-robot interaction," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2419–2425.

[14] I. Ha, Y. Tamura, and H. Asama, "Development of open platform humanoid robot DARwIn-OP," *Advanced Robotics*, vol. 27, no. 3, pp. 223–232, 2013.

[15] M. Lapeyre, S. N'Guyen, A. Le Falher, and P.-Y. Oudeyer, "Rapid morphological exploration with the Poppy humanoid platform," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 959–966.

[16] H. Shi, W. Wang, S. Song, and C. K. Liu, "ToddlerBot: Open-source ML-compatible humanoid platform for loco-manipulation," *arXiv preprint arXiv:2502.00893*, 2025.

[17] Y. Chi, Q. Liao, J. Long, X. Huang, S. Shao, B. Nikolic, Z. Li, and K. Sreenath, "Demonstrating Berkeley Humanoid Lite: An open-source, accessible, and customizable 3D-printed humanoid robot," in *Robotics: Science and Systems XXI*, 2025.

[18] A. Pirrone, "Open Duck Mini: Making a mini version of the BDX droid," GitHub repository, https://github.com/apirrone/Open_Duck_Mini, 2024, repository created May 2024; accessed: Nov. 17, 2026.

[19] 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, vol. 164, 2022, pp. 91–100.

[20] 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.

[21] NVIDIA, M. Mittal, P. Roth, J. Tigue, A. Richard *et al.*, "Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning," *arXiv preprint arXiv:2511.04831*, 2025.

[22] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs, C. Sferrazza, Y. Tassa, and P. Abbeel, "Demonstrating MuJoCo Playground," in *Robotics: Science and Systems XXI*, 2025.

[23] 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*, 2012, pp. 5026–5033.

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

[25] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 143:1–143:14, 2018.

[26] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems XVI*, 2020.

[27] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer, "VMP: Versatile motion priors for robustly tracking motion on physical characters," *Computer Graphics Forum*, vol. 43, no. 8, p. e15175, 2024, Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA).

[28] M. A. Hopkins, G. Wiedebach, K. Cesare, J. Bishop, E. Knoop, and M. Bächer, "Interactive design of stylized walking gaits for robotic characters," *ACM Transactions on Graphics*, vol. 43, no. 4, pp. 1–16, 2024.

[29] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[30] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3803–3810.

[31] 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 (ECC)*, 2019, pp. 3420–3431.