An Embodiment-Adaptive RL Pipeline for Robotics Locomotion with a Morphology Gap*

Jude Onyenze
Department of Mechanical Engineering, University of Texas at Dallas
Richardson, Texas, USA
chisimdi.onyenze@utdallas.edu
ORCID: 0009-0000-1418-1058

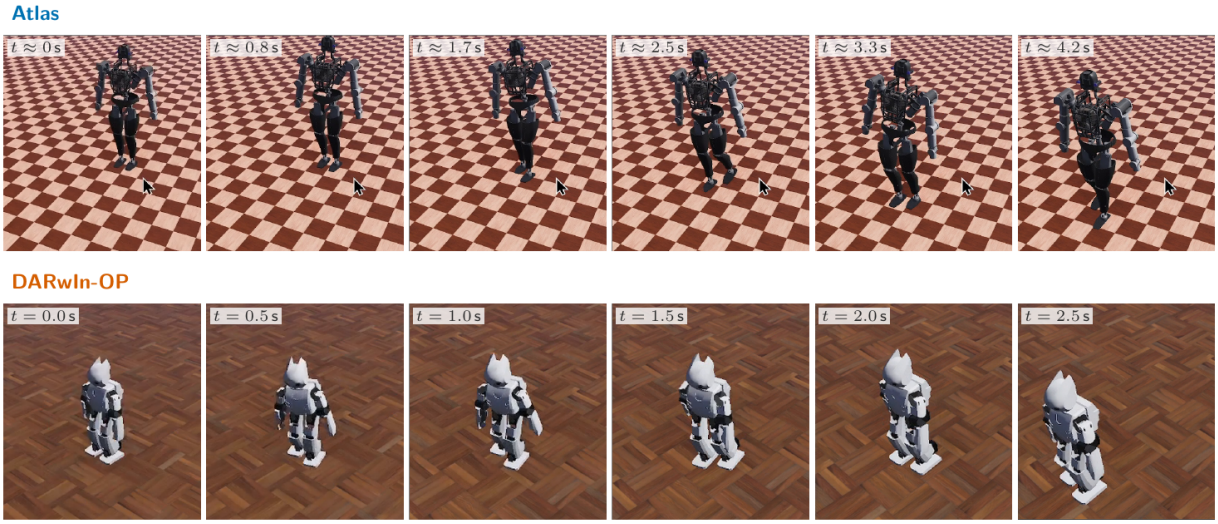


Figure 1: Recorded frames of both robots walking. Top: Atlas, about 0.83 s between frames. Bottom: DARwIn-OP, 0.5 s between frames (cropped around the robot, which walks toward the camera). Frames are cropped from screen captures of the Webots window.

Abstract

Our motivation is cost and access. As GPU prices rise and AI models grow larger and more computationally intensive, training a reinforcement-learning or other learning agent for a new task is becoming harder for anyone without a large compute budget. We ask how far a small controller can get on a CPU alone. We describe a single-file learning controller that uses only the Python standard library, runs on a CPU or on almost any device, and takes a humanoid in the Webots simulator from nothing to a walking gait: it calibrates joint sign conventions, standing height and lean, searches a parameterised central-pattern-generator (CPG) gait, hardens the gait against command noise, and optionally trains a TD3 residual policy on top. We applied the same pipeline to two robots with a large morphology gap, the DARwIn-OP and the Boston

*Project page: https://mro9.github.io/An-Embodiment-Adaptive-RL-Pipeline-for-Robotics-Locomotion-with-a-Morphology-Gap

Dynamics Atlas. Calibration found the correct joint conventions on both robots without hand-tuning; on the Atlas the chosen sign combination scored -0.007 against -0.48 for the nearest alternative and -7 to -9 for the rest. A curriculum (march in place, then stride creep, then search) produced an Atlas gait that covered 3.06 m in 7.7 s (about 0.40 m/s, 23.7 cm steps) and stayed up in 3 of 3 runs under ± 0.003 rad command noise, after 296 candidate gaits (at least 38 simulated minutes; wall-clock time was not logged). The Atlas needed a per-robot profile, search ranges and a hard-robot mode, so we call the method embodiment-adaptive. The TD3 residual did not help on either robot: in a three-hour Atlas run the forward gait stayed up in 19 of 137 forward episodes, all in the first 44 episodes of the run. We report this as a negative result and discuss likely causes.

1 Introduction and Related Work

Getting a new humanoid to walk usually means a hand-built controller, a GPU training stack, or both. We ask how far a small, dependency-free program can get on its own, and what has to be told to it about each robot. The practical motivation is accessibility: the whole pipeline runs on a laptop CPU with only the Python standard library and the Webots controller API.

One line of work trains a single policy that controls many body shapes. URMA, for example, learns one locomotion policy across multiple embodiments (2). Residual learning on top of a fixed controller is established in manipulation (7, 15), and a central pattern generator is a classic compact prior for locomotion (6). Recent work also speeds up off-policy training for humanoids, but on GPUs with parallel simulation: FastTD3 (14) combines parallel simulation, large batches and a distributional critic with TD3, and FlashSAC (8) bounds weight, feature and gradient norms to keep large-scale critics stable. Our setting is the opposite end of that range: one robot, one CPU, and no framework. Our work is different in kind: nothing is shared between robots except the method. Each robot gets its own gait and, if used, its own residual network.

Our contributions are:

- a self-calibration procedure for joint signs, standing height and lean that ran unchanged on two robots of very different size;
- a hard-robot curriculum that avoids a stand-and-sway local optimum on the Atlas;
- an honest account of what had to be set per robot, and of a residual-RL stage that did not help.

1.1 The Embodiment Gap

The two robots differ in size, strength, sensing and dynamics, which is what makes them a useful pair. The Atlas is the Boston Dynamics humanoid, a research platform whose optimisation-based control was described by Kuindersma et al. (10). In the simulator it stands 2.7 times higher at the root (0.921 m against 0.339 m), its leg joints can exert about 104 times the peak torque seen in calibration (260 against 2.5 N·m), and its arm joints about 85 times (170 against 2 N·m). The DARwIn-OP (4) reports joint positions; the Atlas has no position sensors, so the controller treats the commanded angles as the measured ones. Even the hip sign convention differs, and the Atlas needed a hard-robot curriculum that the Darwin did not. Yet the gait periods the pipeline ended up using differ by only 1.4 times (0.64 s against 0.90 s). A possible reading, which we have not tested, is that the gait family transfers while the stability margin does not.

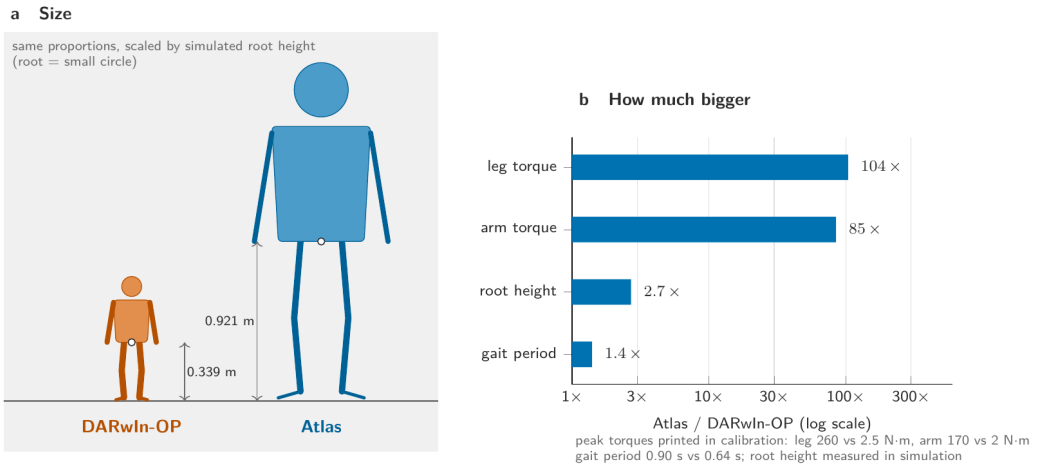


Figure 2: The embodiment gap. (a) The two robots drawn with the same proportions, scaled by their simulated root height (schematic). (b) Atlas divided by DARwIn-OP on a log axis. Torques are the peak values printed during calibration.

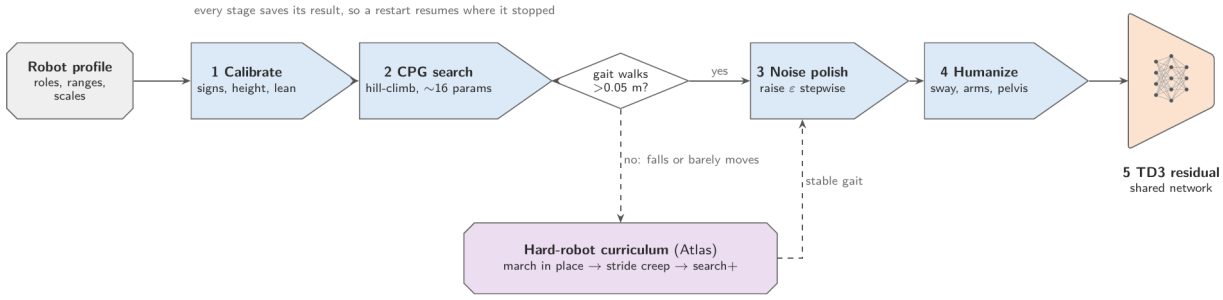


Figure 3: The pipeline. Solid arrows: every robot. Dashed arrows: the hard-robot curriculum, entered when the ordinary search leaves a gait that falls or moves under 0.05 m. Keyboard mode (not drawn) drives the saved CPG gait only.

2 Method

The controller is one Python class run inside the Webots simulator (12). Its stages run in order (Figure 3 and Table 1), and each stage saves its result so a restart resumes where it stopped.

The fitness rewards distance and staying upright and penalises lateral drift and yaw. For hard robots the bonuses are scaled by how long the steps are, which removes the reward for standing and swaying.

A robot profile carries what cannot be measured: joint names and roles, which joints rest or hold a fixed pose, arm targets, a leg-length and time scale, motor speed, a minimum root height, search ranges, noise and action scales. The profile is detected from the device names. If the ordinary search leaves a gait that falls or moves under 0.05 m, the controller switches itself to the hard-robot mode.

Table 1: Pipeline Stages

Stage	**What it does**
Calibration	Tries joint-sign combinations and scores each by how well the robot stands; scans stand lean and crouch; measures the floor height; checks the sign of the gyro/accelerometer-based balance feedback
Gait search	Hill-climbs a CPG gait of about 16 continuous parameters (period, hip, knee and ankle amplitudes, lean, roll, foot ratio, yaw, arm swing, side-step terms and four feedback gains)
Hard-robot curriculum	March in place, then raise hip swing level by level (each level must pass repeated runs and travel a minimum distance), then search from the best stable gait
Noise polish	Raises command noise level by level and keeps a candidate only if it still walks and survives the noise
Humanize	Adds sway, arm swing and pelvis rotation; on hard robots a candidate must pass repeated clean and noisy runs
Residual RL	TD3 with layer normalisation learns a small correction added to the gait, one shared network over the gaits, exploration noise capped by each gait's measured tolerance

2.1 Gait Generator

The gait is a central pattern generator (6), a set of phase-shifted sinusoids that give a target angle for each leg joint. With period T, crouch height c from calibration and a start-up ramp that fades the motion in over 0.6 s, the phase, the ramp and each leg's phase are

$$\phi(t) = \frac{t}{T} \bmod 1, \tag{1}$$

$$\rho(t) = \min\Big(1, \frac{t}{0.6\,\mathrm{s}}\Big), \tag{2}$$

$$\phi_s = \Big(\phi + \frac{s}{2}\Big) \bmod 1, \quad s \in \{0, 1\}. \tag{3}$$

The hip, knee and ankle pitch targets and the hip roll of leg s are

$$h_s = \tfrac{c}{2} - A_h\,\rho\,\cos 2\pi\phi_s, \tag{4}$$

$$k_s = c + A_k\,\rho\,\max\big(0, \sin 2\pi\phi_s\big), \tag{5}$$

$$a_s = \kappa\,(k_s - h_s) + (1-\kappa)\tfrac{c}{2} + \ell_0 + \lambda\,\rho, \tag{6}$$

$$r = A_r\,\rho\,\sin 2\pi(\phi + \phi_r). \tag{7}$$

Here A_h, A_k and A_r are the hip, knee and roll amplitudes, κ is how strongly the ankle follows the leg, ℓ_0 is the calibrated stand lean, λ is a searched lean offset and ϕ_r is the roll phase. The foot roll is μr, with μ the foot ratio, the yaw joints and arms follow the same phase, and each target is multiplied by the joint sign found in calibration. The balance feedback adds terms proportional to the measured pitch, roll and their rates (gains k_p, k_d, k_{roll}, k_{hip}) to the ankle, hip and foot targets. To test robustness, uniform noise of size ε is added to every command:

$$\tilde{q}_j = q_j + \varepsilon\,u_j, \qquad u_j \sim \mathcal{U}(-1, 1). \tag{8}$$

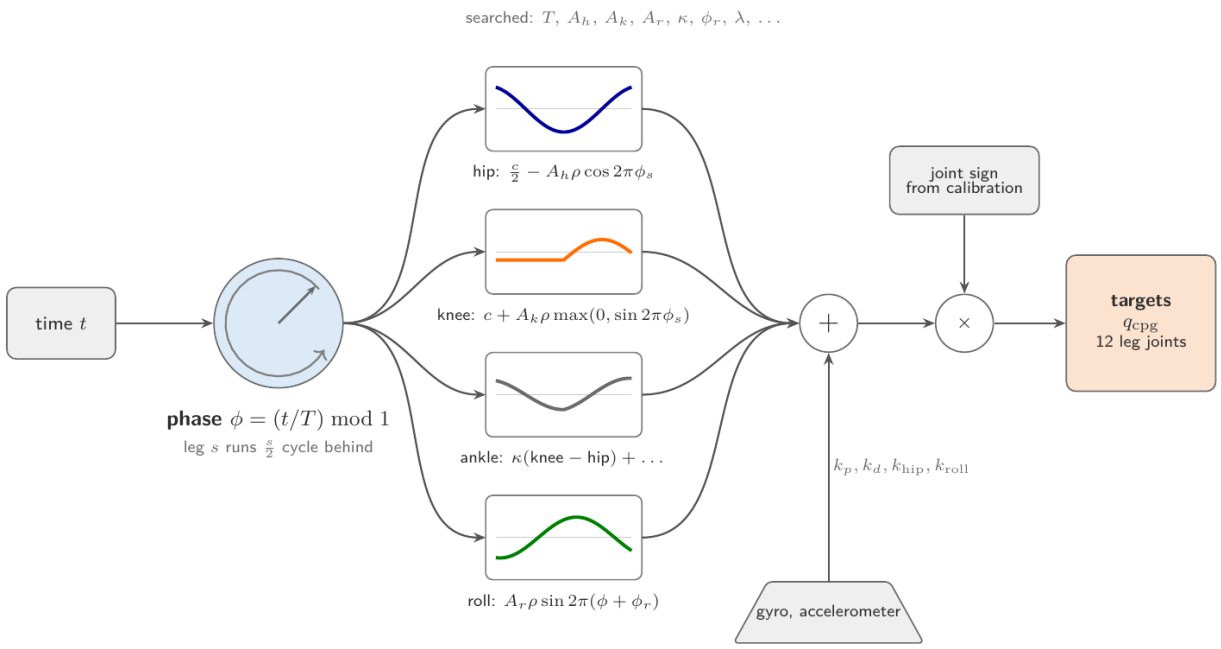


Figure 4: The CPG gait generator. Feed-forward targets come from the phase; the balance feedback terms act on the ankle, hip and foot roll. About 16 parameters are searched; each window sketches the waveform of one joint command.

2.2 Fitness

Each candidate gait is run for 240 control steps (7.7 s). With d the distance travelled in the wanted direction, y the sideways drift, ψ the yaw, n_{alive} the steps before a fall, n the episode length, ℓ the step length (one step is half a cycle) and h the human-likeness score, a gait that stays up scores

$$f_{\text{up}} = d + 0.15\,\frac{n_{\text{alive}}}{n} - 0.8\,|y| - 0.3\,|\psi| + p\,B, \tag{9}$$

$$B = 0.5\,h + 1.4\,\text{clip}\big(\tfrac{\ell}{\ell^*}\big) + 1.2\,\text{clip}\big(\tfrac{A_k}{0.6}\big), \tag{10}$$

where B is the bonus term, $\ell^* = 0.13$ m and clip limits its argument to $[0, 1]$. A gait that falls is ranked by how long it stayed up, never by how far it went:

$$f_{\text{fall}} = -1 + 0.9\,\frac{n_{\text{alive}}}{n} + 0.2\,\text{clip}_{[-0.5,0.5]}(d). \tag{11}$$

The bonus gate p is 1 on the DARwIn-OP. On the Atlas (hard-robot mode) it is $p = \text{clip}(\ell/(\ell^*/2))$, so standing and swaying earn no bonus. Marching in place uses a separate fitness that rewards knee lift and penalises drift. Because a hard-robot gait can pass once and fall the next time, a candidate is also judged by pass counts, $S = (c, m)$: the runs out of n_c clean and n_n noisy runs that stay up.

2.3 Residual Learner

The TD3 (3) residual (7, 15) adds a bounded correction to the gait. The action a_t has 12 components, each limited to $a_{\max} = 0.06$ rad, and the commanded joint targets are

$$q_t = q_{\text{cpg}}(\phi_t) + a_t, \tag{12}$$

$$a_t = a_{\max}\,\mathrm{clip}_{[-1,1]}\big(\pi_\theta(s_t) + \eta_t\big). \tag{13}$$

The state s_t has 48 entries: 12 joint offsets from the standing pose, 12 joint velocities, 3 gyro and 3 accelerometer readings, sine and cosine of the gait phase, the 12 previous actions and a 4-way one-hot of the gait direction, which lets one network serve all four gaits. The reward for a step with forward speed v along the wanted direction, target speed v^*, tilt θ, sideways speed v_l, yaw ψ and penalty weight p (ramped from 0 to 1 over the first 6,000 steps) is

$$r_t = \mathrm{clip}_{[-4,3]}\big(r_t^{\text{track}} - p\,r_t^{\text{pen}} - p\,r_t^{\text{act}}\big), \tag{14}$$

$$r_t^{\text{track}} = 0.4 + 2\,e^{-((v-v^*)/0.06)^2}, \tag{15}$$

$$r_t^{\text{pen}} = 1.5\,\theta^2 + 2\,|v_l| + 0.5\,|\psi|, \tag{16}$$

$$r_t^{\text{act}} = \tfrac{0.1}{12}\left(\left\|\tfrac{a_t}{a_{\max}}\right\|^2 + \left\|\tfrac{a_t - a_{t-1}}{a_{\max}}\right\|^2\right). \tag{17}$$

Training is TD3 with two critics, target networks and delayed actor updates. The target for a transition (s, a, r, s', d), where $d = 1$ only for a real fall, uses target-policy smoothing:

$$y = r + \gamma\,(1-d)\,\min_{i=1,2} Q_i'(s', \tilde{a}), \tag{18}$$

$$\tilde{a} = \mathrm{clip}_{\pm a_{\max}}\big(a_{\max}\,\pi_{\theta'}(s') + \tilde{\xi}\big), \tag{19}$$

$$\tilde{\xi} = \mathrm{clip}_{[-0.3,0.3]}(\xi), \quad \xi \sim \mathcal{N}(0, 0.1^2). \tag{20}$$

The critic loss, the actor gradient and the target-network update are

$$\mathcal{L}(Q_i) = \mathbb{E}\big[(Q_i(s,a) - y)^2\big], \tag{21}$$

$$\nabla_\theta J = \mathbb{E}\big[\nabla_a Q_1(s,a)\,a_{\max}\nabla_\theta \pi_\theta(s)\big], \tag{22}$$

$$\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}. \tag{23}$$

The settings were $\gamma = 0.99$, $\tau = 0.005$, Adam with learning rate 3×10^{-4} for the critics and 1×10^{-4} for the actor, batch size 16, a 50,000-transition replay buffer, one gradient step every 4 control steps once 1,500 transitions are stored, the actor updated every second critic update and held back for the first 1,500 updates. The networks, optimiser and replay buffer are described below. Exploration noise η_t fell from 0.30 to 0.05 (in units of $a_{\max}$) over 30,000 steps and was capped at the gait's measured noise tolerance, halved on hard robots.

2.4 Networks, Optimiser and Replay Buffer

Every network is a multilayer perceptron (MLP) with one hidden layer of 64 units. The hidden pre-activation is normalised across its 64 units (layer normalisation (1), with no learned gain or bias) and passed through a ReLU (13). The actor ends in a tanh, so its 12 outputs lie in $[-1, 1]$ before scaling by $a_{\max}$. Each critic ends in one linear unit and takes the state and action together, $x = [s; a]$, 60 inputs.

$$h = \mathrm{ReLU}\big(\mathrm{LN}(W_1 x + b_1)\big), \tag{24}$$

$$\mathrm{LN}(z)_i = \frac{z_i - \mu_z}{\sqrt{\sigma_z^2 + 10^{-5}}}, \tag{25}$$

$$\pi_\theta(s) = \tanh(W_2 h + b_2), \quad Q(s,a) = W_2 h + b_2. \tag{26}$$

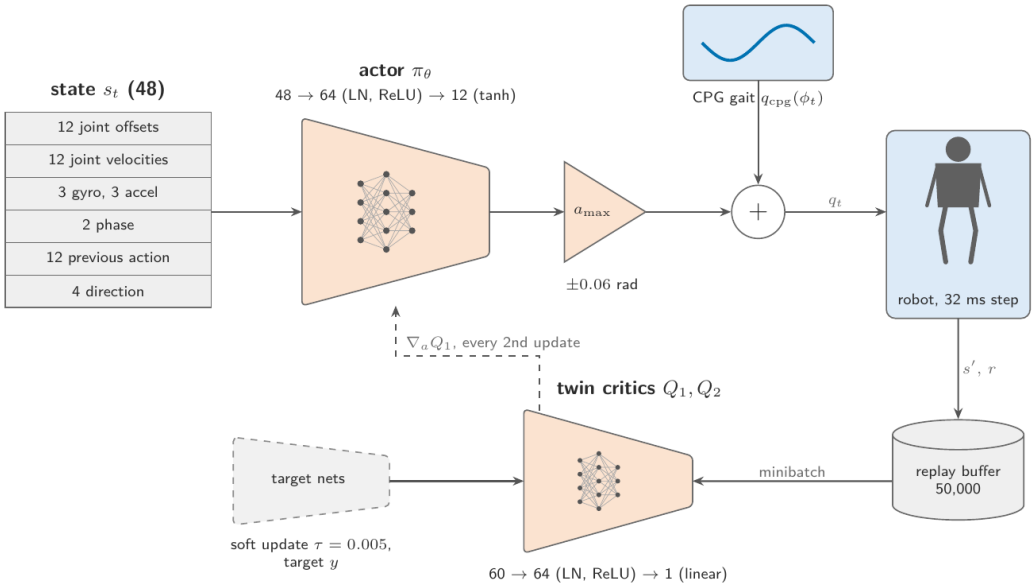


Figure 5: The TD3 residual learner. The actor outputs a bounded correction, at most 0.06 rad per joint, that is added to the CPG joint targets. LN = layer normalisation; each network has one hidden layer of 64 ReLU units. The critics see the state and the action (60 inputs). Trapezoids are neural networks, the cylinder is the replay buffer.

Weights start from a Gaussian with variance 2 divided by the fan-in (5). The output layers are scaled by 0.1, so the untrained actor gives a correction near zero and the robot begins as the plain gait. The actor has 3,916 parameters and each critic 3,969, which is 11,854 trainable parameters in all, plus the target copies. No machine-learning framework is used: the forward and backward passes, layer normalisation and the optimiser are written out in plain Python lists, which is what keeps the program to the standard library.

The optimiser is Adam (9) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Each gradient is averaged over the minibatch and clipped elementwise to $[-5, 5]$ before the step. The critics train alone for the first 1,500 updates, so they learn what a fall costs before the policy starts to move.

The replay buffer (11) is a first-in first-out ring of 50,000 transitions $(s_t, a_t, \tilde{r}_t, s_{t+1}, d_t)$. When it is full the oldest transition is overwritten, and minibatches of 16 are drawn uniformly. Because one network serves all four gaits, the buffer mixes transitions from every direction; for the first 6,000 steps only the forward gait is sampled. The terminal flag d_t is 1 only when the robot falls, never when an episode times out. A fall also costs 3 reward points, and rewards are divided by the running standard deviation of the discounted return (never by less than 1) so the critic sees values of a stable size:

$$r'_t = r_t - 3 \cdot \mathbb{1}[\text{fall}], \quad G_t = \gamma G_{t-1} + r'_t, \tag{27}$$

$$\tilde{r}_t = \frac{r'_t}{\max(\sigma_G,\, 1)}. \tag{28}$$

Because the program runs on one CPU thread, learning is throttled: no update happens until 1,500 transitions are stored, and after that one gradient step is taken every 4 control steps, an update-to-data ratio of 0.25. Exploration noise is held for several steps at a time (noise repetition): the hold length k is drawn with probability proportional to $k^{-1.5}$, up to 8 steps, so most holds are short and a few are long. Figure 6 shows the network, the optimiser and the replay buffer together.

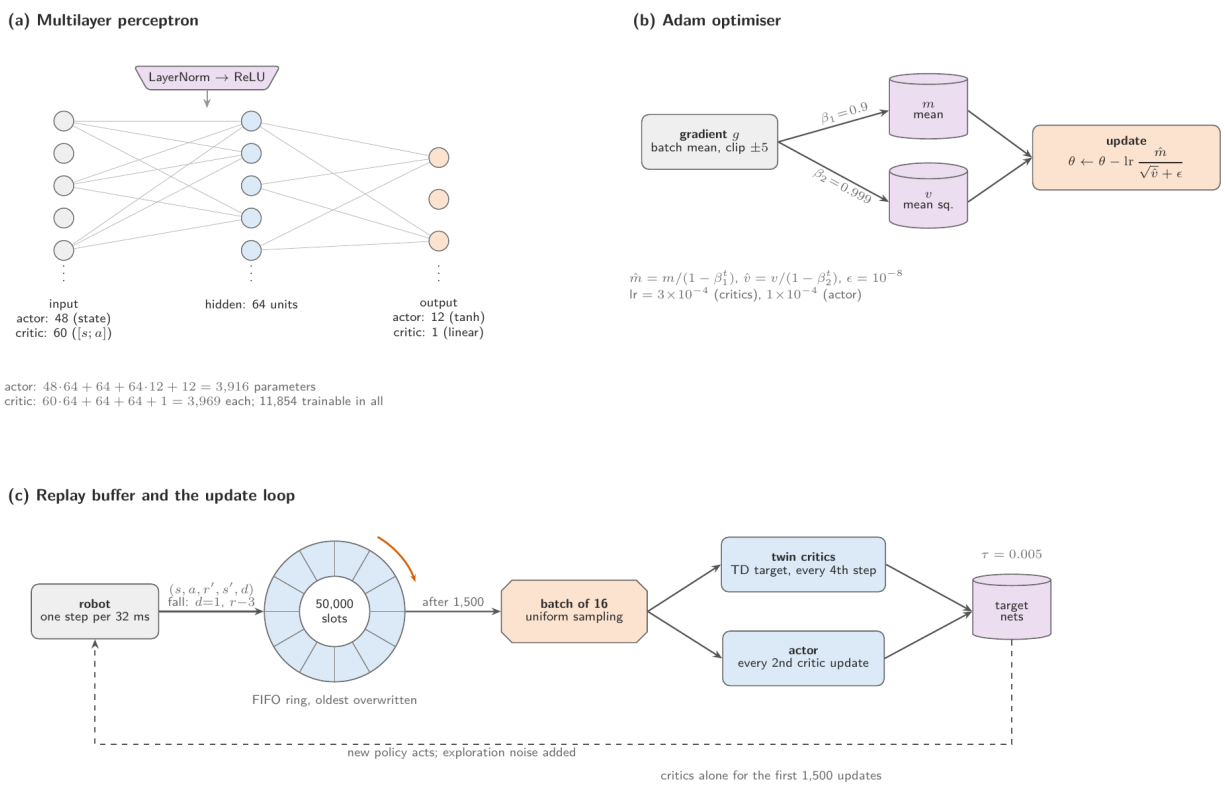


Figure 6: The learning machinery in one view. (a) The multilayer perceptron: one hidden layer of 64 units with layer normalisation and ReLU; the actor maps the 48-entry state to 12 tanh outputs, each critic maps the 60-entry state and action to one linear output. (b) Adam keeps a running mean m of the gradient and of its square v, corrects both for their start-up bias, and scales each step by the ratio. (c) Each control step the robot writes one transition into a 50,000-slot ring buffer; after 1,500 transitions a batch of 16 is drawn uniformly every fourth step, the twin critics learn alone for the first 1,500 updates, the actor updates on every second critic update, and the target networks follow with $\tau = 0.005$. Figure 5 shows what the actor's output does to the gait; this figure shows how it is trained. Not to scale.

3 Experimental Setup

All experiments ran in Webots R2025a (12) with a 32 ms control step, on a CPU, in Python with no third-party packages. An episode is 300 control steps (9.6 s) and ends early on a fall. Table 2 lists the two robots.

We report distance travelled, whether the robot stayed up, step length, and, for hard robots, pass counts over repeated runs, because the same Atlas gait can pass once and fall the next time. Robustness is the number of runs, out of 3, that stay up under random command noise of a given size.

Table 2: The Two Robots in the Simulator

	DARwIn-OP	**Atlas**
Standing root height	0.339 m	0.921 m
Peak leg torque (calibration)	2.5 N·m	260 N·m (hip)
Joint position sensors	yes	no (commanded angles used)
Profile length / time scale	1.0 / 1.0	2.6 / 1.6

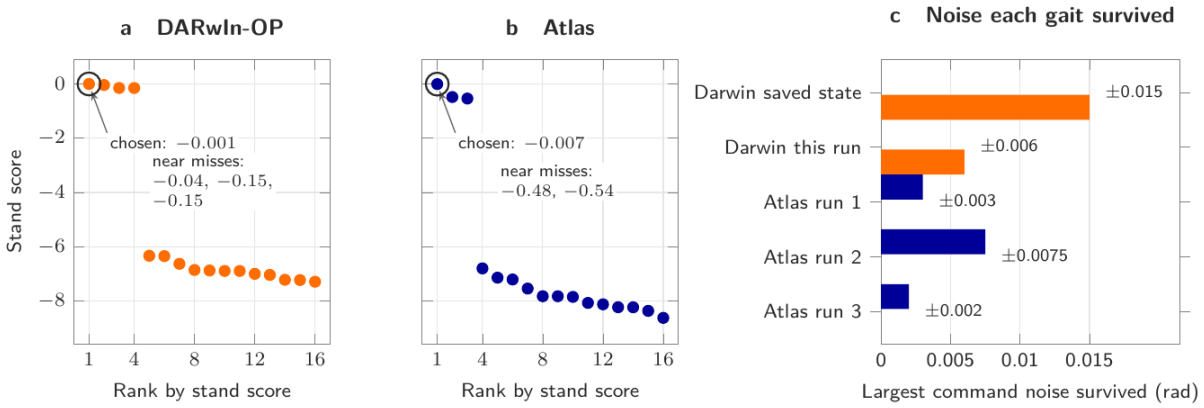


Figure 7: (a, b) Stand scores of all 16 joint-sign combinations on each robot, ranked (chosen one circled). Both robots have a few near-miss combinations that also stand, so the score alone does not always single out one answer; the log adds a joint-limit hint to break ties. (c) The largest command noise each saved or found gait survived. The Atlas tolerance varied between runs from ±0.002 to ±0.0075 rad, against ±0.006 to ±0.015 rad for the Darwin.

4 Results

4.1 Calibration

On the Atlas the same sign combination (hip $[-1,-1]$, knee $[1,1]$, ankle $[-1,-1]$) was chosen on every run; the chosen combination scored -0.007 in the third run (-0.006 in the first), the two nearest alternatives -0.48 and -0.54, and the rest -7 to -9. Standing height was 0.921 m with tilt 0.01. The stable stand-lean window was about -0.02 to $+0.05$ rad. On the DARwIn-OP calibration chose hip $[-1,1]$, knee $[1,-1]$, ankle $[1,-1]$ and a standing height of 0.339 m, again matching its saved state (Figure 7).

4.2 Atlas Gait Search (One Logged Run)

Table 3 summarises the stages of the logged run.

A second from-scratch run found a different gait (period 1.096 s against 1.192 s) with a measured noise tolerance of ±0.0075 rad, and a third found one that walked 3.89 m in 9.6 s (0.405 m/s, 18.3 cm steps) with a tolerance of ±0.002 rad. Independent runs therefore reach different solutions of different robustness.

Table 3: Atlas Gait Search, One Logged Run

Stage	Result
Blind search without the curriculum	stand-and-sway shuffle, about 2 cm steps, 0 of 3 noisy runs survived
March in place	found at iteration 24, fitness +0.975
Stride levels 0.02 to 0.13	passed; step length rose from 7.7 cm to 20.3 cm (0.13 passed 3 of 4 runs); 0.16 failed
Search from the best stable gait	3.06 m in 7.7 s (about 0.40 m/s), 23.7 cm steps, 9.1 cm sway; re-run 3.84 m in 9.6 s; stayed up in 3 of 3 runs at ±0.003 rad

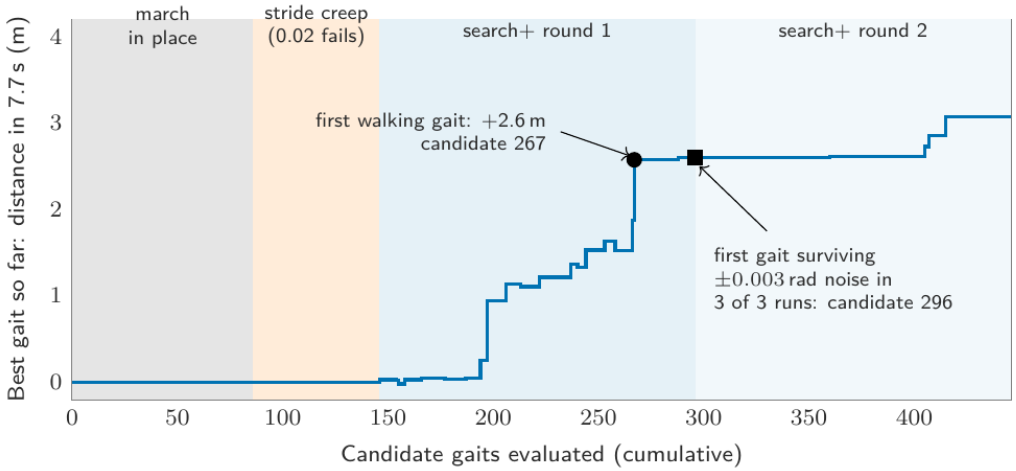


Figure 8: Distance of the best Atlas gait so far against cumulative candidates evaluated (third run). Shaded bands are pipeline stages; the march stage is about 86 candidates (16 seeds plus 70), stride creep 60, search 150 each.

4.3 How Long the Atlas Needed to Reach a Stable Gait

In the third run's console the first 146 candidate evaluations (marching in place, then two stride-creep retunes) produced no forward motion. Search from the best stable gait then climbed from about 0 m to +2.6 m in 7.7 s within roughly 120 candidates, and a gait that also stayed up in 3 of 3 runs under ±0.003 rad noise was confirmed after 296 candidates (Figure 8). At 7.7 s of simulated time per candidate that is at least 38 simulated minutes; wall-clock time was not logged. About 43 minutes of training to the best result were reported for an earlier Atlas run. The DARwIn-OP's equivalent curve is missing: its walking gait was restored from an earlier run whose search was not logged in the supplied consoles.

4.4 Atlas Residual RL

The shared network was trained for 444 episodes and 10,066 updates, about three hours (Figure 9). All 19 forward episodes that stayed up (mean +3.83 m) came from the first 44 episodes of the run. The back, left and right gaits had all failed their open-loop checks before training, and the

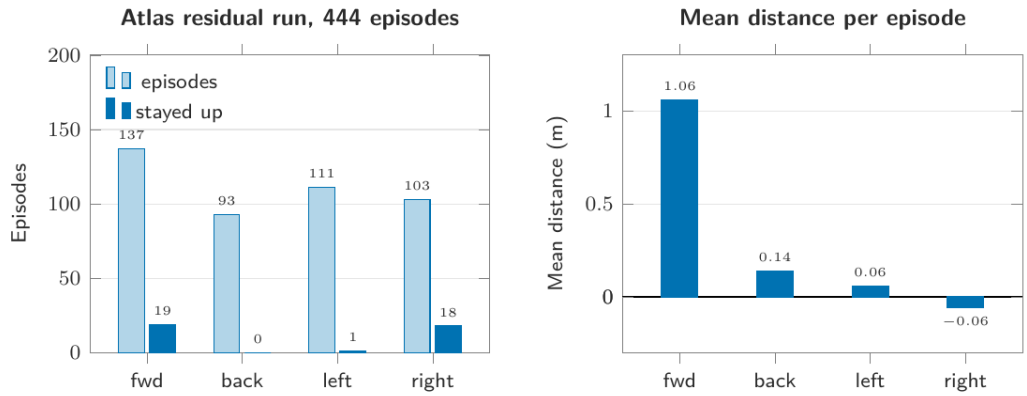


Figure 9: The Atlas residual run by gait direction. Left: episodes run and episodes that stayed up. Right: mean distance per episode. The 18 right-gait episodes that stayed up travelled about 0 m.

right-gait episodes that stayed up travelled about 0 m.

4.5 DARwIn-OP From-Scratch Run

The run lasted 165 episodes. The restored gait fell after 4.4 s in the first check, which switched the controller into the hard-robot mode. Noise polish measured ±0.006 rad. Open-loop, the left and right gaits walked +1.35 m and +1.95 m in 9.6 s without falling; the forward gait moved +0.618 m in 6.7 s and fell. Under residual training the forward gait stayed up in 8 of its first 35 episodes and 1 of the next 36; the best 10-episode mean distance (+1.077 m, episode 44) was reached when left and right episodes were being sampled, and it fell to about zero by episode 160 (Figure 10).

4.6 Commanded Joint Motion

The logs do not record measured joint positions, and the Atlas has no position sensors, so Figure 11 shows the commanded targets of the two saved forward gaits, computed from the gait parameters (feed-forward part only; the balance feedback terms and the per-joint sign are left out). In radians the hip and ankle commands of the two robots are of similar size, but the Darwin lifts its knee higher above the crouch (about 0.30 against 0.17 rad) and completes a cycle in 0.64 s against 0.90 s for the Atlas.

4.7 Humanize Values

The humanize stage adds sway, arm swing and pelvis rotation (Figure 12). On the Atlas, run A raised sway from 8.6 to 11.4 cm and the human-likeness score from 0.53 to 0.54, but that gait fell on re-run; run B accepted no candidate (score 0.60 before and after). The Atlas gait swings its arms about twice as far as the Darwin's in radians (0.56 against 0.27). No humanize log for the Darwin was supplied.

4.8 Screen-Recorded Atlas Walking

Figure 13 shows frames from a screen recording of the Atlas in keyboard-control mode, which drives the saved CPG gait without the residual network. The robot walks toward the camera with

alternating steps and arm swing. The same 86 s recording also contains several falls, so it shows what the gait looks like, not how reliable it is. No usable Darwin footage was available for this figure: the two Darwin clips supplied are 0.3 s and 0.57 s long and show the robot standing during residual training.

4.9 Side-by-Side Recordings

Figure 1 compares recorded frames of the two robots walking. The Atlas row comes from the keyboard-control recording. The DARwIn-OP row comes from a separate recording made during the noise-polish stage, where the Darwin walks its saved gait toward the camera. The two recordings were made in different runs with different cameras, so the rows show motion style, not matched conditions.

4.10 Side-by-Side Gait Sketch

Figure 14 puts the two saved forward gaits next to each other. It is computed, not recorded: thigh and shank angles come from each gait's commanded hip and knee targets (feed-forward part only), with the same assumed crouch of 0.30 rad for both robots, and both robots are drawn at the same size. Because both use the same gait family, the sketched motions look alike. They differ mainly in amplitude (hip swing 0.19 against 0.14 rad, knee lift 0.30 against 0.17 rad) and in cycle time (0.64 s against 0.90 s). Real Darwin footage would turn this into a recorded comparison.

5 Discussion

Adaptive. The calibration, gait family, fitness structure, noise polish and residual learner carried over without redesign. The Atlas still needed a profile (joint roles, arm targets, scales), its own search ranges and seed gaits, balance feedback, a hard-robot curriculum, and reduced noise and action scales.

Why the residual did not help (hypotheses, not tested). Three things look likely from the logs. First, the shared network trained on gaits that did not walk; the sideways and backward gaits mostly fell or stood still. Second, standing still paid: Atlas "right" and Darwin "back" episodes stayed up with about 0 m travelled and returns above 150. Third, exploration noise was capped at the measured tolerance, and at that level about every second Atlas forward episode fell. We changed the code to leave failing gaits out of training and to halve the exploration cap, but have run that for only six episodes, so no effect is claimed. A forward-only run with and without the residual is the missing ablation.

Fragile search results. Hard-robot gaits did not repeat exactly, and the first humanize stage accepted a single lucky run that then fell. Judging gaits by pass counts over repeated clean and noisy runs fixed this, at the cost of more simulation time per candidate.

6 Limitations and Reproducibility

- Simulation only, one simulator, two robots; no hardware.
- Few runs per robot and no seeds are reported; results from different controller versions were logged into the same files, so the Darwin logs are not a single controlled experiment.

- No baselines (hand-tuned or learned) and no ablations.
- The escalation to the hard-robot mode fired on the Darwin from a single fall of a gait that mostly walks.
- The Atlas has no joint position sensors, so commanded angles stand in for measured ones.
- Omni-directional gaits are weak on both robots.
- The method is adaptive, not general: each robot needed its own profile, search ranges, seed gaits, balance feedback, a hard-robot curriculum and reduced noise and action scales. Further Atlas gains looked like they would need enough robot-specific code that the method could no longer honestly be called general, which is why we stopped.

The controller is one Python file for Webots R2025a. The Atlas world needs the robot's `supervisor` field set to TRUE. State, model and log files are written next to the controller and are named per robot. The code is available at https://github.com/MrO9/An-Embodiment-Adaptive-RL-Pipeline-for-Robotics-Locomotion-with-a-Morphology-Gap. The exact configuration of each logged run is not yet included there.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[2] N. Bohlinger et al. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. arXiv preprint arXiv:2409.06366, 2024. Conference on Robot Learning (CoRL) 2024.

[3] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, 2018.

[4] I. Ha, Y. Tamura, and H. Asama. Development of open platform humanoid robot DARwIn-OP. *Advanced Robotics*, 2013. doi: 10.1080/01691864.2012.754079.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[6] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4), 2008.

[7] T. Johannink et al. Residual reinforcement learning for robot control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[8] D. Kim et al. FlashSAC: Fast and stable off-policy reinforcement learning for high-dimensional robot control. arXiv preprint arXiv:2604.04539, 2026.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[10] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3), 2016. doi: 10.1007/s10514-015-9479-3.

[11] L.-J. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8, 1992.

[12] O. Michel. Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 2004.

[13] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[14] Y. Seo, C. Sferrazza, H. Geng, M. Nauman, Z.-H. Yin, and P. Abbeel. FastTD3: Simple, fast, and capable reinforcement learning for humanoid control. arXiv preprint arXiv:2505.22642, 2025.

[15] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling. Residual policy learning. arXiv preprint arXiv:1812.06298, 2018.

A Further Method Details

The Method section covers the gait generator, the fitness and the residual learner. Four further parts of the controller carry much of the adaptation to an unknown body and are described here with the equations they use: calibration and the standing pose, the balance feedback, the gait search with its noise polish, and the march and stride-creep curriculum used on hard robots. All constants are those in the controller source; none is a measured result.

A.1 Calibration and the Standing Pose

The controller is given no sign map between its joint convention and the robot's. For each of the three leg joints (hip pitch, knee, ankle pitch) and each leg it tries both signs. A joint-limit prior fixes a sign first where the two joint ranges are lopsided (range ratio above 1.4), which shrinks the search. Each remaining combination is judged by a short standing trial of 40 ticks (about 1.3 s): the robot is reset to the standing pose and must hold it with the command signs applied (Figure 15). With tilt θ_t, height drop $\Delta_t = z_0 - z_t$, the mean taken over the last half of the trial, $\Delta_{\max} = \max_t \Delta_t$ and $\theta_{\max} = \max_t \theta_t$:

$$S = -\bar{\theta}_{\text{tail}} - 2\max(0, \Delta_{\max}) - 5\,\mathbb{1}[\theta_{\max} > 1], \tag{29}$$

$$b = 0.3\, n_{\text{hip}} + 0.15\,\mathbb{1}[\sigma_{a,L} = -\sigma_{a,R}]. \tag{30}$$

Here n_{hip} counts the hip signs that agree with the prior, and b is a small hint that mirrored legs usually need opposite ankle signs. The chosen map maximises $S+b$. A second scan then picks the standing posture. Some robots are heavier at the back, so an upright crouch can tip over. The scan tries crouch $c \in \{0.30, 0.15\}$ and a constant forward lean $\ell \in \{-0.08, -0.04, 0, 0.04, 0.08, 0.12, 0.16, 0.20\}$ rad, using 70-tick trials, and prefers the smaller lean on ties. It stops at the first crouch whose best score exceeds -0.2:

$$(c^\star, \ell^\star) = \arg\max_{c,\ell} \Big[S(c, \ell) - 0.01\,|\ell| \Big]. \tag{31}$$

A.2 Balance Feedback and the State Estimator

The feedback gains in the gait generator act on a body-pitch and body-roll estimate taken from the supervisor orientation matrix R (row-major, entries R_6 and R_7). Because a robot's sign conventions are not known in advance, a sign s_{fb} is calibrated first: the standing robot is leaned with the ankle command by $+0.12$ and -0.12 rad for 20 ticks each, and the measured pitch is compared. If the pitch change δ is below -0.004 rad the sign is flipped; if $|\delta|$ is smaller than 0.004 the default is kept.

$$\theta_p = -s_{\mathrm{fb}} \arcsin R_6, \quad \theta_r = \arcsin R_7, \tag{32}$$

$$\dot{\theta}_p \leftarrow 0.5\,\dot{\theta}_p + 0.5\,\frac{\theta_p - \theta_p^{\mathrm{prev}}}{\Delta t}, \tag{33}$$

$$\delta = \hat{\theta}_p(+0.12) - \hat{\theta}_p(-0.12). \tag{34}$$

The rates are smoothed with weight 0.5 (the previous rate and the new finite difference count equally). The corrections added to the gait targets are scaled by the start-up ramp ρ, so they fade in with the walking motion:

$$\Delta q_{\mathrm{hip}} = \rho\, k_{\mathrm{hip}}(\theta_p + 0.1\,\dot{\theta}_p), \tag{35}$$

$$\Delta q_{\mathrm{ankle}} = -\rho\,(k_p\,\theta_p + k_d\,\dot{\theta}_p), \tag{36}$$

$$\Delta q_{\mathrm{foot}} = \rho\, s_r\, k_{\mathrm{roll}}(\theta_r + 0.1\,\dot{\theta}_r). \tag{37}$$

Side-stepping adds a lateral term to the same phase. With the leg phase ϕ_l, side amplitude A_s and side phase ϕ_s the roll joint is driven by $s_r(\mathrm{roll} + \mathrm{lat})$, where $\mathrm{lat} = A_s \rho \cos 2\pi(\phi_l + \phi_s)$, and the foot roll by $s_r(\mu\,\mathrm{roll} + k_s\,\mathrm{lat})$. Backward walking runs the phase in reverse, and turning scales the hip amplitude. The six directions therefore share one parameterised gait.

A.3 Gait Search and the Noise Polish

The hill-climb (Figure 16) mutates a parameter vector p with ranges r_i. Each parameter is perturbed with probability 0.5, sign-flip parameters flip with probability 0.08, and values are clamped to their ranges (the roll phase wraps modulo 1). The step size σ adapts every ten tries:

$$p_i \leftarrow p_i + \mathcal{N}\big(0, (\sigma r_i)^2\big), \tag{38}$$

$$\sigma \leftarrow \mathrm{clip}_{[0.03,\,0.4]}(g\,\sigma). \tag{39}$$

where $g = 0.8$ if fewer than two of the last ten candidates were accepted and $g = 1.25$ otherwise. A candidate is scored under command noise η (uniform, added to every leg joint), taking the worse of two runs so that a gait that works only in one exact run scores badly. It replaces the best gait only if it still beats it on a repeat run, and on hard robots in the humanize stage also only if it stays up in two of two clean and two of two noisy runs:

$$f_{\mathrm{rob}}(g) = \min\big(f(g;\eta),\, f'(g;\eta)\big), \tag{40}$$

$$f_{\mathrm{keep}} = \min\big(f_{\mathrm{rob}},\, f'_{\mathrm{rob}}\big) > f^{\star} + 10^{-6}. \tag{41}$$

The human-likeness score H in the fitness is a weighted sum of four clipped terms: lateral sway w (half the range of the sideways motion after removing a straight-line drift, target 2.5 cm), arm swing, pelvis yaw and swing-phase knee bend. With weights $\omega = (0.3, 0.3, 0.2, 0.2)$, inputs

$x = (w, |A_{\text{arm}}|, |A_{\text{yaw}}|, A_k)$ and scales $x^* = (0.025, 0.7, 0.08, 0.6)$, and with the step length ℓ measured per half cycle,

$$H = \sum_{i=1}^{4} \omega_i \,\text{clip}\big(x_i / x_i^*\big), \tag{42}$$

and

$$\ell = \frac{\max(0, d)}{n_{\text{alive}}\,\Delta t \,/\, (T/2)}. \tag{43}$$

The noise polish (lower half of Figure 16) then raises the noise level through $\eta \in \{0.003, 0.006, 0.010, 0.015\}$ rad. At each level the current gait is tested; if it fails, 30 candidates are searched at $\sigma = 0.10$. On the Atlas a candidate is kept only if at least two of three clean and two of three noisy runs stay up and its mean distance is at least $0.6\,d_0$, where d_0 is the distance of the unpolished gait. The first level that cannot be passed ends the polish, and the last level passed is the noise tolerance used later to cap RL exploration. A candidate that walks much less far than the original (under $0.6\,d_0$) is never kept.

A.4 The March and Stride-Creep Curriculum

On hard robots a random gait almost always falls, so a hill-climb has no signal to follow. The controller therefore first learns to march in place, where balance with one foot in the air is the only problem, and only then grows the stride (Figure 17). The march searches a restricted set of parameters (period, knee lift, ankle follow, lean, roll amplitude and phase, foot ratio and the feedback gains) from a pool of 16 seeds (two roll-mirror settings, four roll phases, two roll amplitudes) with hip swing, yaw and arm swing set to zero, $T = 1.3$ s and knee lift 0.30 rad. Its fitness rewards staying up, lifting the knees, and not drifting; f_{fell} applies to a gait that falls and f_{up} to one that stays up:

$$f_{\text{fell}}^{\text{march}} = -1 + 0.9\,\tfrac{n_{\text{alive}}}{n}, \tag{44}$$

$$f_{\text{up}}^{\text{march}} = 0.4 + 0.8\,\text{clip}\big(\tfrac{A_k}{0.30}\big) - 0.8\,(|d| + |y|) - 0.3\,|\psi|. \tag{45}$$

The march is accepted when its fitness reaches 0.9 and, to end the search early, when it reaches 1.0 and two noisy runs stay up. The hip swing A_h is then raised through eight levels (0.02, 0.04, 0.06, 0.08, 0.10, 0.13, 0.16, 0.20 rad). A level passes only if the gait stays up and really travels. With c clean and m noisy passes out of two runs each and mean distance d over the passing clean runs:

$$\text{pass} \iff (c + m \geq 3) \wedge (c \geq 1) \wedge \big(d \geq \min(0.4,\, 6A_h)\big). \tag{46}$$

A failing level re-tunes only the balance parameters (up to two rounds of 30 candidates) before it is judged again, and the curriculum stops at the first level that still fails. The gait is saved after every level, so an interrupted run resumes from the largest good swing.

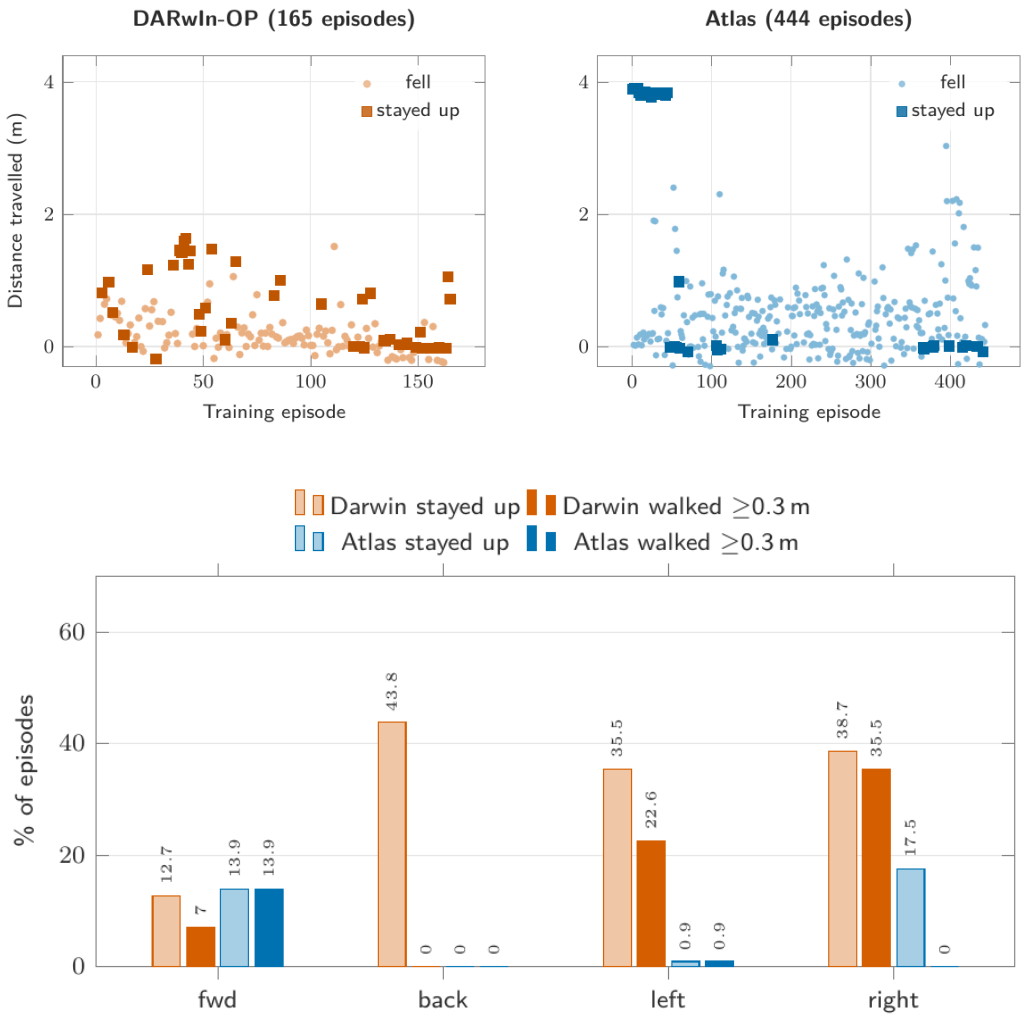


Figure 10: Residual training on both robots, typeset from the logged episodes. (a) Distance travelled in every episode; squares stayed up, dots fell. The Atlas squares at about 3.9 m are the first 19 forward episodes; after that almost nothing stays up and moves. (b) Share of episodes by gait direction that stayed up (pale) and walked at least 0.3 m (solid). Only 14% of Atlas forward and 7% of Darwin forward episodes walked; the Darwin's left and right gaits walked in 23% and 35%, while the Atlas's back, left and right gaits stood or fell.

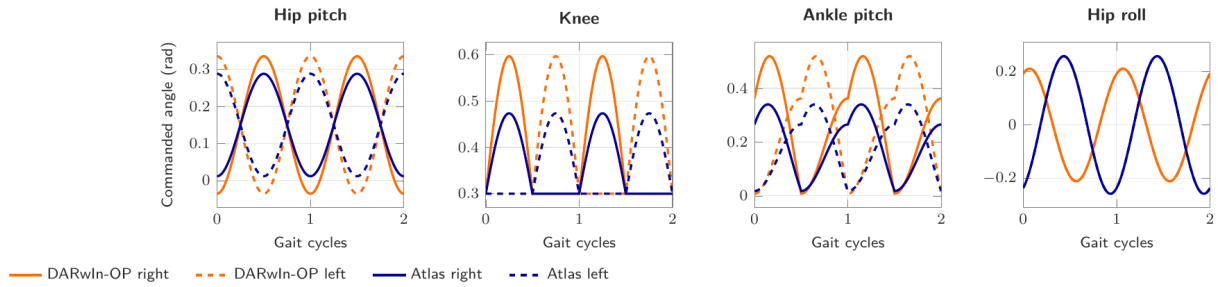


Figure 11: Commanded hip pitch, knee, ankle pitch and hip roll for two cycles of each saved forward gait (DARwIn-OP: restored gait; Atlas: third-run search gait). Solid lines are the right leg, dashed lines the left. The horizontal axis is gait cycles: one cycle is 0.64 s for the DARwIn-OP and 0.90 s for the Atlas.

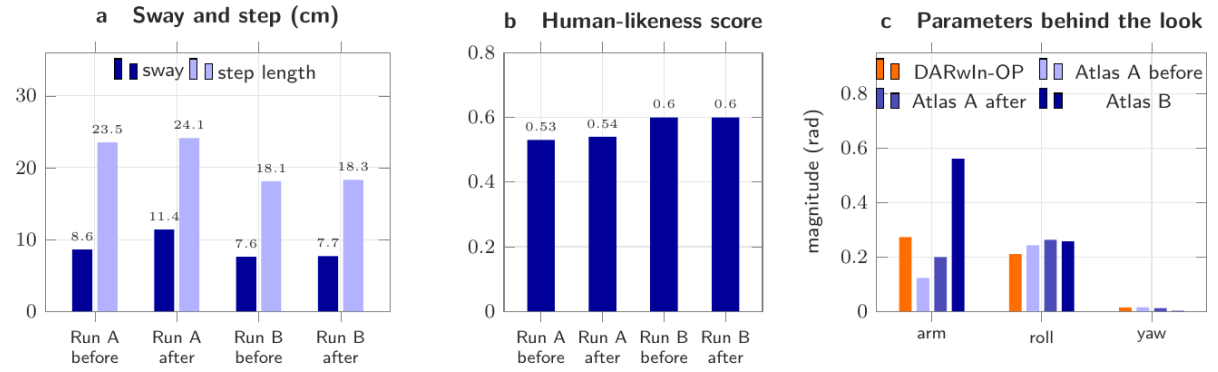


Figure 12: Humanize values. (a) Sway and step length and (b) human-likeness score before and after the stage on two Atlas runs; run A's humanized gait fell on re-run and run B accepted no candidate. (c) The gait parameters that set the humanized look, with the DARwIn-OP gait for reference.

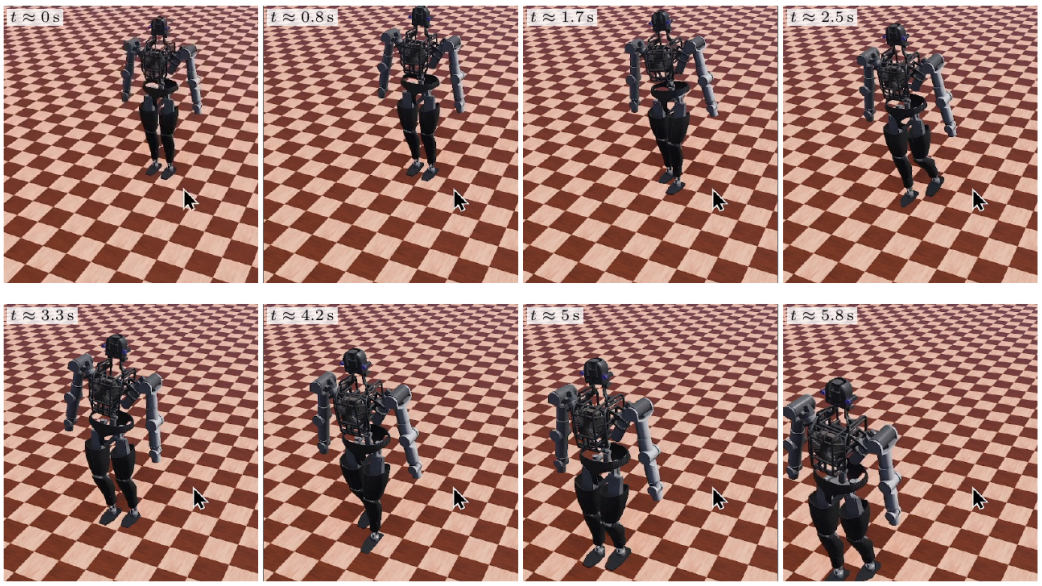


Figure 13: Eight frames, about 0.83 s apart, from the 86 s keyboard-control recording of the Atlas (the walking stretch starting at 10 s in the recording; t is time from the first frame shown). Frames are cropped from a screen capture of the Webots window.

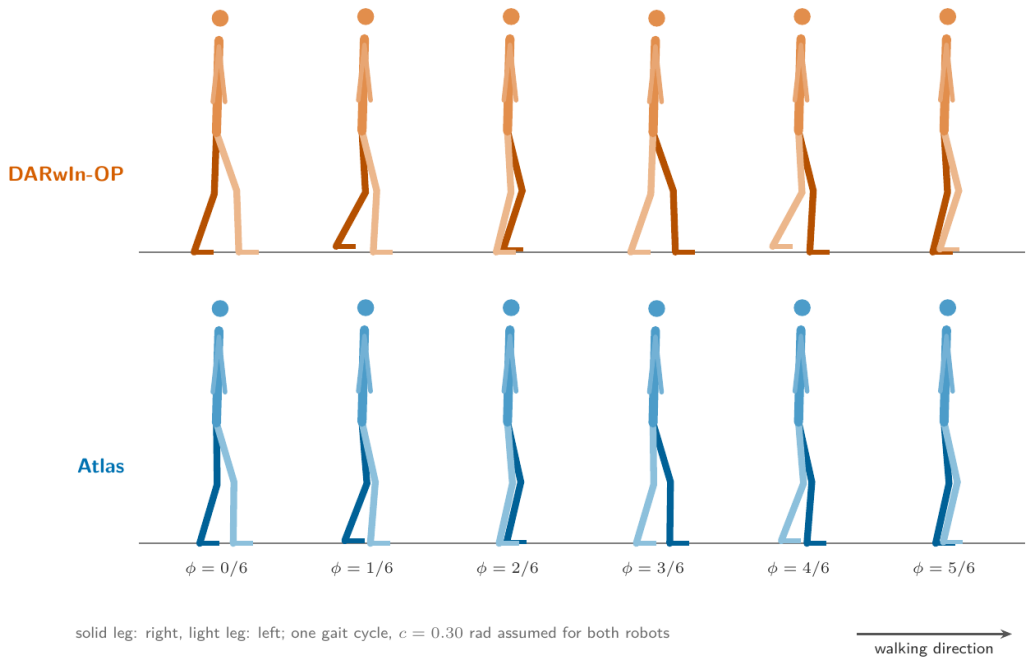


Figure 14: One gait cycle of the saved forward gait of each robot at six phases, drawn from the commanded targets (a sketch, not simulator output). Solid leg: right; light leg: left. Arms are drawn at rest.

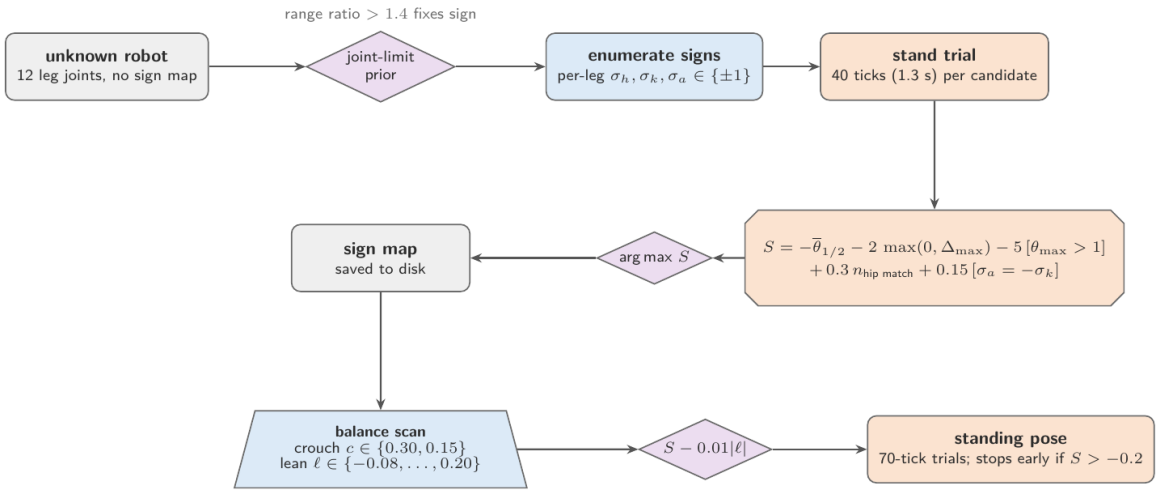


Figure 15: Calibration. A joint-limit prior and an enumeration of per-leg signs feed a short standing trial scored by S; the arg max gives the saved sign map, and a balance scan over crouch and lean gives the standing pose.

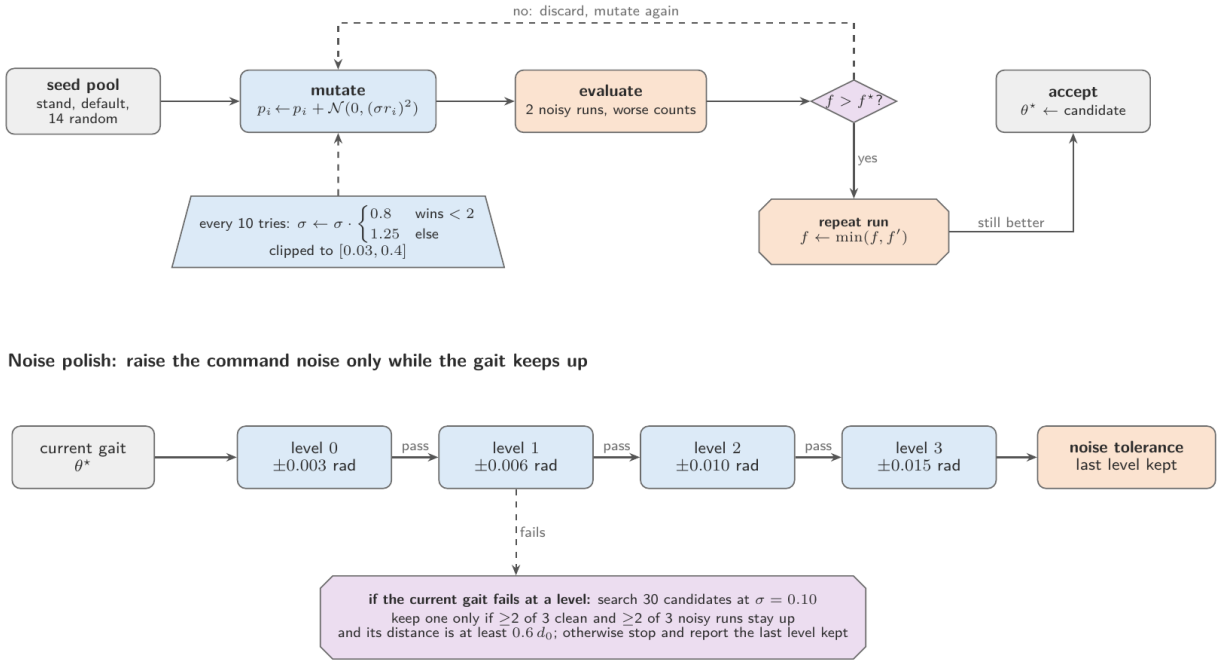


Figure 16: Top: the hill-climb loop with its repeat-run check and step-size adaptation. Bottom: the polish ladder. A failed level triggers a short search; a candidate that does not meet the pass counts is rejected and the polish stops.

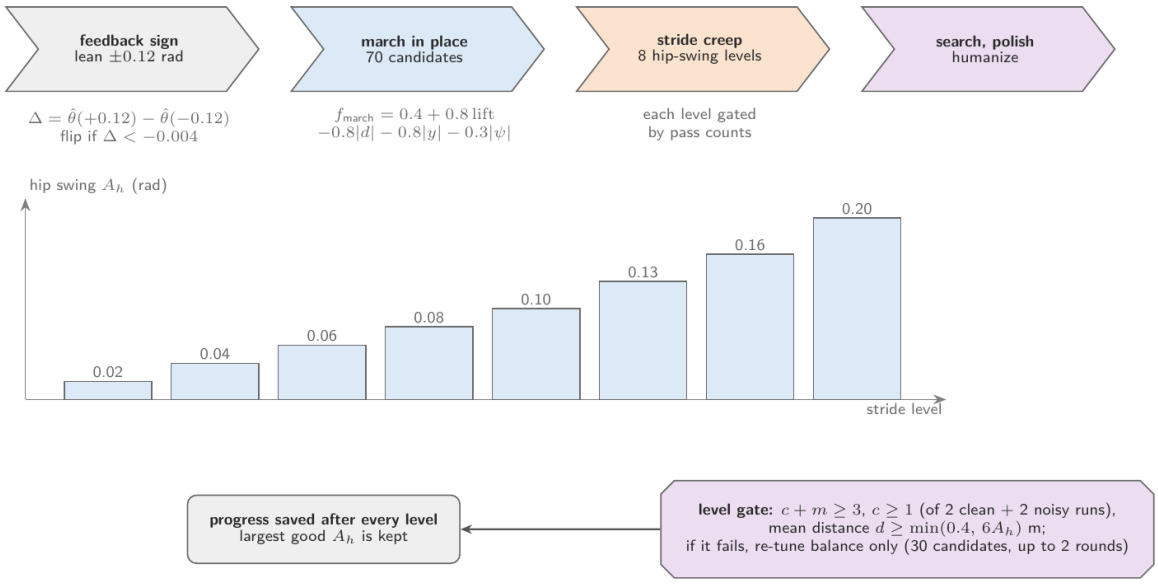


Figure 17: The curriculum for hard robots: a feedback-sign check, a march in place, eight stride levels each behind a pass-count gate, and then the normal search, polish and humanize stages.