Electronic Research Archive

ERA, 34(3): 1524–1545.
DOI: 10.3934/era.2026069
Received: 11 November 2025
Revised: 02 February 2026
Accepted: 04 February 2026
Published: 13 February 2026

http://www.aimspress.com/journal/era

Research article

Human-like arm swing strategies in ES–SAC humanoid gait: Stability and performance on flat vs rough terrain

Mustafa Ayyıldız[1,*] and Övünç Polat[2]

[1] The Vocational School of Technical Sciences, Department of Electronics and Automation, Akdeniz University, Antalya, Turkey
[2] Faculty of Engineering, Department of Electrical and Electronics Engineering, Akdeniz University, Antalya, Turkey

* **Correspondence:** Email: mustafaayyildiz@akdeniz.edu.tr; Tel: +902423106700.

Abstract: This study presents a hybrid evolution strategy–soft actor-critic (ES–SAC) controller for a reduced-degrees of freedom spatial (3D) humanoid gait model with predominantly sagittal-plane motion that focuses on arm-leg coordination. Policies are trained on flat terrain only and then evaluated on both flat and uneven ground under identical simulation settings. Arm-leg coordination is examined systematically in three modes (counter-phase with the legs [normal], in-phase [anti-normal], and fixed [passive]), and the results are compared with findings from human experiments. Whereas most prior studies evaluate policies primarily via reward curves, this work conducts a deep analysis using interpretable metrics aligned with human-like walking: speed-normalized power and torque, lateral and vertical deviations, and moment-balance terms. Simulation outcomes are reported quantitatively through these metrics rather than reward alone. Across five random seeds, a clear terrain-dependent trade-off appears between the swinging strategies: anti-normal attains higher forward speed and lower torque-per-speed, whereas normal provides better lateral tracking and lower power-per-speed on rough ground. Directional trends agree with human experiments (e.g., immobilized or in-phase arms raise metabolic cost), while numerical gaps reflect that the simulator measures mechanical power rather than metabolic energy. Within this framework, the impact of coordinated arm swing on balance and efficiency is quantified with a breadth and clarity uncommon in the literature.

Keywords: humanoid robot gait; arm swing strategies; evolutionary computation; actor-critic algorithm; uneven terrain; reinforcement learning

1. Introduction

Humans typically swing their arms when walking. Arm swing requires muscle activity and can be energetically costly. It is unclear whether potential benefits elsewhere in the body compensate for this cost. Humanoid robot designs likewise draw inspiration from human gait [1]. However, most approaches to humanoid walking control [2–5] have focused on leg motion, placing the contribution of the upper body and arm joints in the background. This gap is particularly important because arm coordination can simultaneously influence whole-body balance regulation, kinematic tracking, and energy-related effort, which are core determinants of practical humanoid locomotion.

In this study, the aim was to examine in detail how arm swing influences the walking performance of humanoid robots using a deep reinforcement learning (DRL)–based controller. To this end, three arm-model scenarios were compared. In the first, the arms were completely fixed with no swing allowed. In the second, the arms swung in-phase with the ipsilateral legs (for example, the right arm moves forward as the right leg moves forward). In the third, the arms swung in counter-phase, synchronised with the contralateral legs as in natural human gait (for example, the left arm swings forward when the right leg steps forward). Across these three conditions, several measures determining gait stability, energy expenditure, and balance were analysed. In this way, the effects of arm motion on bipedal gait dynamics were quantified with numerical evidence. In addition, the comparison is designed to reveal the relative advantages and disadvantages of each arm strategy in terms of stability–speed behaviour and effort or economy, rather than treating arm motion as a purely aesthetic degree of freedom. Because the counter-phase pattern resembles typical human coordination while the in-phase pattern does not, the analysis also highlights where humanoid control behaviour aligns with, or departs from, human-like gait coordination.

According to Collins et al. [1], arm swing during walking serves important functions. They showed that fixing the arms increases the metabolic energy cost of walking by about 12%. When the arms swing in-phase with the legs on the same side, energy use can rise by up to 26%. Another notable finding is that arm swing markedly reduces the vertical-axis ground-reaction moment at foot-ground contact. When the arms are completely fixed, this moment increases by about 63% compared with normal swing. The shoulder torque required for normal arm swing is also very small, indicating that with only a minor muscular cost the arms contribute both to balance and to the energetic efficiency of walking. At the same time, transferring these findings to humanoid robots is not straightforward, because robot morphology, actuation, and controller objectives can shift the trade-offs between human-like coordination and task-optimal coordination.

DRL methods have produced satisfactory results for humanoid walking [6–8]. Recent studies show that using the arms can markedly improve walking speed and balance. In particular, at higher walking speeds the swing of the legs can generate unbalanced angular momentum, making the robot prone to stumbling [9,10]. As a remedy, Zhang et al. [9] integrated an angular-momentum minimisation term into the reward of a DRL-based controller, enabling the robot to use its arms actively to counteract yaw-plane sway and stabilise the gait.

Prior studies [1,11,12] suggest that arm swing can reduce centre-of-mass motion in the vertical plane, balance whole-body angular momentum, and lower the ground reaction moment about the vertical axis. When the arms move out of phase with the legs, they counter the angular momentum generated by the legs, thereby reducing uncontrolled trunk rotations and excessive lateral sway. These insights motivate a controlled comparison of arm coordination strategies in a humanoid robot,

using metrics that reflect both kinematic performance and effort or energy usage, rather than relying only on reward curves. Accordingly, the central questions are whether arm coordination improves or degrades steady-state gait quality, energy-related cost proxies, and robustness-related balance indicators under a fixed controller and training setup, and which strategy best mediates the trade-offs among these objectives.

A single agent architecture is used that combines evolution strategies (ES) with soft actor-critic (SAC) into a hybrid ES–SAC algorithm. The ES and SAC components are developed with inspiration from [13,14] and [6–8], respectively. With this agent, both leg and arm motions are controlled, and the reward function is designed accordingly as a multi-component objective that targets forward progression and fall avoidance while explicitly encouraging stability- and efficiency-related behaviour. Beyond learning curves, a detailed steady-state gait analysis is performed to interpret how arm coordination changes walking dynamics in a measurable way. In parallel, the effect of arm-swing modelling on the learning process itself is investigated by examining how each arm strategy shapes training behaviour under ES–SAC, including convergence tendencies, variability across seeds, and value-related learning signals such as Q0. In addition, a multi-seed design with five independent runs per condition is adopted to reduce run-specific fluctuations and to assess stability and generalisability more reliably. Overall, by bringing biomechanical findings together with ES–SAC control and by comparing fixed, in-phase, and counter-phase arm strategies under the same training setting, a structured assessment is provided of how arm motion shapes the trade-offs among gait stability, balance, and energy-related performance in humanoid locomotion.

2. Humanoid model

The humanoid model used in this study is simulated as a spatial (3D) multibody system; however, its joint structure is deliberately simplified and designed with a low number of degrees of freedom (DoF). This simplification enabled faster learning for the ES–SAC model. Each leg consists of three single-axis revolute joints (hip–knee–ankle). The anatomical correspondence of the joint axes is defined through fixed transform/alignment blocks within the joint subsystems. Accordingly, the lower extremities provide a total of 6 DoF (3 DoF per leg), and all of these joints are directly actuated via torque inputs. For the upper extremities, each arm is represented as a 3-joint kinematic chain; however, the wrist and elbow joints are fixed (passive/locked), and only the shoulder joints are torque-controlled. Thus, although the arms include rigid links connected in 3D space, active control is applied only at the shoulder; the arm-swing effect is examined through this single active shoulder DoF. Under this setup, the model has 8 active (controlled) DoF in total: 6 DoF from the legs and 2 DoF from the shoulders.

Joint ranges of motion were constrained using angular limits (joint limits) to remain consistent with human anatomy. The hip joint was limited to a range of -90° to +30°, allowing the leg to swing within a prescribed envelope relative to the torso while preventing excessive angular deviations. The knee joint was constrained to 5°–90°; selecting a lower bound greater than zero avoids full "locking" (full extension), which is intended to yield numerically more stable contact dynamics and more realistic step-to-step transitions. The ankle joint was limited to -20° to +20°, providing a safety margin against instabilities, such as excessive plantarflexion/dorsiflexion during foot-ground contact and abrupt growth of contact forces [15,16]. For the upper extremity, the shoulder sagittal motion was constrained to -30° to +90°. This keeps arm swing within a biomechanically plausible range and prevents extreme arm postures that could be dynamically unstable or inconsistent with the intended

arm-swing behaviour. Since the remaining joints in the arm chain (e.g., elbow and wrist) were modelled as passive/locked, these shoulder limits effectively define the active motion envelope of the upper extremity and thus serve as the primary kinematic constraint for the arms. Taken together, these constraints aim to keep the model within realistic joint motion ranges while maintaining a numerically stable simulation regime for contact and torso dynamics, thereby also facilitating faster learning of the ES–SAC controller.

3. ES–SAC model

3.1. Architecture of the ES algorithm

The ES algorithm used in this study employs a population-based search to optimise the parameters of the policy network. At each generation, a population is created from the current policy parameters, and each individual is evaluated by running the agent in the environment and recording the return achieved [17].

ES defines a Gaussian distribution around the actor network's parameters. In every generation, a noise vector is sampled from this distribution and added to the actor parameters. These perturbations enable exploration of a wide range of policies in the search space. In effect, the actor of each population member is a small, randomised variation of the SAC agent's actor network [17,18].

At the end of each generation, the cumulative returns obtained by the population members are compared. These returns act as fitness values for that generation. By averaging over multiple independent trials, noise and stochastic environment effects are smoothed, giving a more reliable estimate of each policy's true performance. A subset of the highest-scoring policies is then selected as elites. This ensures the survival of the best in the evolutionary search [13,14].

The parameters of the elite individuals are used to form the next generation's policy. The base parameters of the SAC actor (i.e., the mean of the Gaussian) are shifted towards the elites, steering the search. Large perturbations are used at the beginning of training to encourage exploration, and the distribution is gradually narrowed near the end to drive convergence to a solution [13,14].

3.2. Hybrid integration of ES and SAC

SAC comprises an actor (policy) network and two critic (Q-value) networks. Its aim is to train the policy to maximise both the cumulative return and the policy entropy. High entropy promotes exploration, while the return reflects the reward obtained from the environment. Each interaction experience is stored in a replay buffer and sampled in mini-batches for gradient-based updates, and target networks are used to improve stability. Combined with ES, which searches in parameter space, SAC learns in data space; in the hybrid ES–SAC algorithm the agent is trained with two learning mechanisms at once [19–21].

First, using the current SAC actor parameters, a population of fixed size is generated. The Gaussian sampling and elite selection described in the previous section are then applied, so the ES-based update acts on the actor. The critic networks are not updated during the ES step. The actor parameters are perturbed as in Eq (3.1):

$$\theta_i = \theta_c + \in_i, where \in_i \backsim \mathcal{N}(0, \Sigma) \tag{3.1}$$

Here, θ_i denotes the actor-network parameters of the i-th individual, θ_c denotes those of the current SAC agent, and $\in_i$ is zero-mean Gaussian noise with covariance matrix Σ.

One of the key advantages of this hybrid architecture is the use of a Replay Buffer. All experiences generated from the population's interaction with the environment (s_t, a_t, r_t, s_{t+1}) are stored in the Replay Buffer. Here, s_t denotes the state, a_t the action, r_t the reward, and s_{t+1} the next state. Both the perturbed data and the data collected by the SAC agent are retained in the buffer. Thus, evolutionary search not only discovers the best individual but also leverages the experience of the entire population, improving efficiency [20].

Once each generation is completed and the actor parameters have been updated within the population by ES, the hybrid algorithm proceeds to the SAC phase, which uses gradient-based learning. Using experiences sampled from the replay buffer (s_i, a_i, r_i, s_{i+1}), each critic network Q_{ϕ_k} is updated by minimising the squared loss L_k against the target value, as in Eq (3.2):

$$L_k = \frac{1}{2M}\sum_{i=1}^{M}\left(r_i + \gamma\left[\min_{j=1,2} Q_{\bar{\phi}_j}(s'_i, a'i) - \alpha \log \pi\,\theta(\,a'_i \mid s'i\,)\right] - Q\phi_k(s_i, a_i)\right)^2 \tag{3.2}$$

Here, M is the mini-batch size, Q_{ϕ_k} is the current critic network and predicts the expected return for a state–action pair, $Q_{\bar{\phi}_K}$ is the target critic, and ϕ_k denotes the learnable parameters of the critic. When computing the target Q-value, the minimum of these networks' estimates is selected, and the result is multiplied by the discount factor (γ) to obtain the present value of future rewards. The term $\alpha \log \pi\,\theta(\,a'_i \mid s'_i\,)$ represents the policy entropy. By weighting the stochastic action distribution with the entropy coefficient (α), the agent's exploration is enhanced. The target critic parameters are then updated using the correction factor (τ), as in Eq (3.3):

$$\overline{\phi_k} = \tau\,\phi_k + (1-\tau)\,\overline{\phi_k} \tag{3.3}$$

After the critics are updated, the actor network is improved in the gradient direction. The actor update proceeds under the guidance of the critics' values, as in Eq (3.4):

$$J_\pi(\theta) = \frac{1}{M}\sum_{i=1}^{M}\left(\alpha \log \pi_\theta\,(\,a'i \mid s_i\,) - \min_{j=1,2} Q_{\phi_j}(s_i, a'_i)\right) \tag{3.4}$$

Here, the actor training improves the policy while maintaining the entropy–reward balance. This update also provides an opportunity to refine the actor parameters after the ES step. If the ES step places the policy in a favourable region, the SAC-based actor update performs fine-grained adjustments. If ES noise deteriorates the policy, the critics' estimates guide the SAC policy update to steer the actor back toward a suitable region [22,23].

A distinctive element of SAC is the dynamic adjustment of the entropy coefficient, which is obtained as in Eq (3.5):

$$L_\alpha = \frac{1}{M}\sum_{i=1}^{M}\left(-\alpha(\log \pi_\theta\,(\,a_i \mid s_i\,) + \bar{\mathcal{H}})\right) \tag{3.5}$$

Here, $\bar{\mathcal{H}}$ is the target entropy. The entropy coefficient α is adjusted to minimise this objective, thereby preserving the desired level of stochasticity.

To summarise, the mathematical steps defined in SAC (the critic losses, the actor objective, entropy tuning, and target-network updates) constitute the algorithm's gradient-based learning stage. The ES–SAC algorithm applies these key steps in alternation by combining them with evolutionary, population-based search steps [17,19,22,24].

3.3. SAC policy architecture

In this study, the agent's observation vector is condition dependent. For the normal and anti-normal configurations, the observation vector is defined as $o_t \in R^{37}$ and decomposed into two components. The first component, $s_t \in R^{29}$, represents the robot's instantaneous state (i.e., the core physical variables required for gait control), while the second component, $u_{t-1} \in R^{8}$, contains the torque commands applied at the previous control step. The motivation for this decomposition is to make the agent's input closer to the Markov assumption in contact-driven locomotion, which is highly dynamic and sensitive to actuation/feedback delays. For the fixed-arms configuration, the observation vector has 31 elements. It includes 25 instantaneous state variables and an additional six-dimensional one-step memory term from the z^{-1} block (i.e., $o_t = [s_t; u_{t-1}] \in R^{31}$ with $s_t \in R^{25}$ and $u_{t-1} \in R^6$). The term u_{t-1} is feedback through a one-sample memory (unit delay, z^{-1}) in the model, helping the agent capture the temporal continuity of torques across consecutive time steps. Within the sensor-based state (s_t), the right and left foot contact indicators enable the agent to distinguish gait phases (e.g., single vs. double support). The torso orientation components (q_x, q_y, q_z) and angular velocities (w_x, w_y, w_z) provide critical feedback for balance maintenance and for early detection of rotational instabilities that may precede falls. The torso lateral position (x), vertical displacement (dz), and linear velocity components (v_x, v_y, v_z) directly encode forward progression, lateral deviation, and vertical oscillations induced by ground contact. Finally, joint angles and angular velocities describe the motion states of the leg joints (hip, knee, and ankle) and the shoulders, enabling the agent to evaluate both step geometry and the effect of arm swing on torso dynamics within a unified observation space.

The actor network used in this study is designed, as shown in Figure 1(a), to capture both the humanoid's instantaneous state and the temporal dynamics of gait. The network input is a 37-dimensional state vector comprising joint angles, velocities, and the system's kinematic properties. These signals are first processed by two fully connected (FC) layers with 256 neurons each, followed by rectified linear unit (ReLU) activations, to yield more abstract and informative representations. In this way, salient features that explain the walking behaviour are extracted from raw observations. The resulting representation is passed to a long short-term memory (LSTM) layer with 256 hidden units, enabling the network to account for not only the current state but also recent steps. The LSTM output then splits into two heads that produce the action mean and the uncertainty of the action distribution, respectively. Expressed as a Gaussian policy, this provides actions that are both stable and exploratory.

The critic network is shown in Figure 1(b). It estimates the value of state-action pairs by jointly evaluating the robot's current state and the action produced by the actor. The first input is a 37-dimensional state vector, processed by a FC layer with 256 neurons. The second input is the actor's 8-dimensional action vector, which is likewise passed through a 256-neuron FC layer. The resulting representations are merged at a concatenation layer to form a shared feature space that captures both state and action information. These fused features are then fed through an additional FC layer with ReLU activations and an LSTM with 256 hidden units, yielding higher-level representations that account for temporal context. Finally, a single-neuron output layer computes the expected value

function $Q(s,a)$ for the state–action pair, supplying the core learning signal that guides the actor's policy updates.

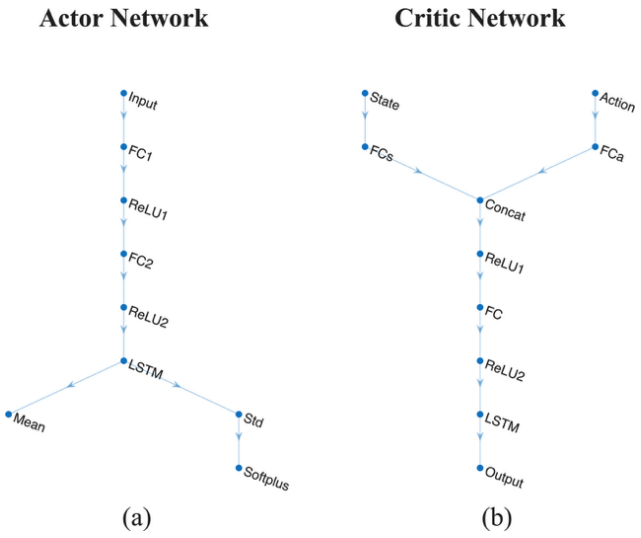


Figure 1. Architectures of the actor (a) and critic (b) networks.

3.4. SAC reward function

In this study, the reward function designed to control the humanoid's gait is multi-component and centres on leg-driven walking performance. Accordingly, it is formulated to prioritise forward progression and continuity of gait. Forward speed and the duration of balanced upright stance are defined as high-priority reward terms. Lateral deviation of the trunk and vertical descent towards the ground plane receive strong penalties. An additional penalty promotes energy efficiency. In this way, the policy is encouraged to maximise stable forward locomotion [8]. In addition, to make arm swing more biomechanically natural, two arm-motion reward components are integrated. These arm terms are assigned small weights so that they act as fine-tuning, improving balance without disrupting walking performance. In sum, the reward function preserves a leg-centred, stable humanoid gait while also encouraging human-like arm–leg coordination.

The total reward is defined as a weighted sum of distinct sub-components. Eq (3.6) shows the components of the reward function:

$$r = w_d r_d + w_f r_f + w_l r_l + w_v r_v + w_p r_p + w_{a1} r_{a1} + w_{a2} r_{a2} \tag{3.6}$$

Here, r_d is the duration reward. At each time step, a fixed positive reward is given as long as the robot remains upright. r_f is the forward-motion reward. r_l is the lateral deviation penalty, a numerical penalty applied to minimise left–right (x-axis) deviations. r_v is the vertical-motion penalty, designed to suppress oscillations along the vertical (z) axis. r_p is the power-consumption penalty. r_{a1} encourages the natural counter-phase swing between the opposite arm and leg. r_{a2} is the arm-symmetry penalty, encouraging the two arms to move symmetrically in counter-phase. The weighting coefficients in the reward are denoted by w. For the Fixed-arms condition, the arm-related terms are removed from the reward function, and the shoulder joints are locked with zero shoulder torques, so no arm motion is present during training and evaluation. Table 1 values were derived after training,

based on simulation observations aggregated across all seeds; where ranges are reported, they denote the minimum–maximum values observed across seeds after convergence. When setting the weights, both the importance of each reward and the numerical range of each reward component were considered, and the coefficients were chosen normalised to these scales. This normalisation is intended to balance the influence of all components on the total reward. Accordingly, by assigning an appropriate weight to each term with reference to the maximum within its natural range, no single component overwhelms the others, and the reward function fulfils its multi-objective aims in a balanced manner. The value ranges of the reward function terms, before multiplication by their gains, are listed in Table 1.

Table 1. Value ranges of reward/penalty terms.

Reward/penalty term	Value range
Duration reward	0.025
Forward velocity reward	-0.5 to 2.5
Lateral penalty	0 to 0.025
Vertical penalty	0 to 0.05
Power penalty	-1000 to 4000
Arm-leg antiphase swing reward/penalty	-0.5 to 1.5
Arm swing symmetry penalty	0 to 0.8

Table 2 lists the reward weights used at the same values for all agents. The duration reward adds a small positive contribution at every step in which the robot sustains walking, encouraging episodes to last as long as possible. The forward-progress reward promotes the primary task (moving forward) by rewarding forward speed; as the table shows, this term can become negative, so backward slip is mildly penalised. The lateral and vertical penalties aim to maintain balance and near-straight travel. The lateral-deviation penalty limits unnecessary side-to-side motion, helping the agent reach the target more quickly, while the vertical-oscillation penalty suppresses jumping or excessive up-and-down body motion. The power penalty enforces energy efficiency by penalising instantaneous torque use across all active joints. The last two terms (arm–leg asynchrony reward/penalty and arm-symmetry penalty) encourage arm motions that resemble natural human gait [8,25,26].

Table 2. Weight values for reward and penalty terms.

Reward weights	Value
Duration reward (w_d)	1
Forward reward (w_f)	2
Lateral penalty, (w_l)	-5
Vertical penalty (w_v)	-10
Power penalty (w_P)	-0.0005
Arm-leg asynchrony (w_{a1})	1
Arm symmetry penalty (w_{a2})	-1

Opposite-phase asynchrony term rewards opposite-side arm and leg motions in opposite directions. It is calculated mathematically as in Eq (3.7):

$$r_{a1} = \theta_{Rs}\theta_{Lh} + \theta_{Ls}\theta_{Rh} \tag{3.7}$$

Here, the right arm angle θ_{Rs} and the left hip angle θ_{Lh}, and the left arm angle θ_{Ls} and the right hip angle θ_{Rh} are denoted. The term in Eq (3.7) is constructed to promote the normal (contralateral, opposite-phase) arm–leg coordination: when the arms swing in opposite phase with the contralateral hips, the summed quantity in Eq (3.7) tends to take positive values, thereby increasing the reward. For the anti-normal (ipsilateral, in-phase) condition, the phase definition is implemented directly inside the reward-signal block by applying a sign convention to one side of the coupled signals in Eq (3.7) before evaluation. Specifically, the phase reference is implemented by negating the hip-angle signals in Eq (3.7) prior to evaluation (i.e., the terms $-\theta_{Lh}$ and $-\theta_{Rh}$ are used in the arm–hip cross-coupling). This operation corresponds to a sign-reference (phase) convention and does not introduce an additional penalty or alter the underlying objective function. In the implementation, hip-angle signals are first sign-corrected (multiplied by -1) to align the forward-swing direction convention across limbs; this preprocessing is applied consistently and does not change the objective, only the sign reference of the angle signals. With this convention, Eq (3.7) remains a consistent coordination indicator whose positive contribution corresponds to the intended Anti-normal mode, rather than being interpreted as an intrinsic penalty against in-phase behaviour.

The second arm reward term, the arm-swing symmetry penalty, evaluates the symmetry of the two arm motions as in Eq (3.8):

$$r_{a2} = (\theta_{Rs} + \theta_{Ls})^2 \tag{3.8}$$

When the arms move in the same direction, the value of the sum becomes large and a high penalty is produced. By contrast, when the arms swing symmetrically in opposite directions, the sum of the two arm angles approaches zero and this term attains its minimum. That is, when the phase difference is 180°, the penalty is minimal. Thus, the robot's arms exhibit a natural counter-swing during walking, with one arm moving forwards while the other moves backwards [1,25].

Table 3. Reward function definitions for the three arm-swing conditions.

Condition	Reward function
Normal	$r = 2\dot{y} - 5x^2 - 10\dot{z}^2 + 0.025 - 0.0005\,P + \theta_{Rs}\theta_{Lh} + \theta_{Ls}\theta_{Rh} + (\theta_{Rs} + \theta_{Ls})^2$
Anti-normal	$r = 2\dot{y} - 5x^2 - 10\dot{z}^2 + 0.025 - 0.0005\,P + \theta_{Rs}(-\theta_{Lh}) + \theta_{Ls}(-\theta_{Rh}) + (\theta_{Rs} + \theta_{Ls})^2$
Fixed	$r = 2\dot{y} - 5x^2 - 10\dot{z}^2 + 0.025 - 0.0005\,P$

**Note*: The minus sign in the anti-normal arm–hip cross-term arises solely from the sign flip applied to the hip-angle signals (θ_{Lh}, θ_{Rh}) to enforce an in-phase coordination convention; it is not an additional penalty term nor a different objective formulation.

Table 3 summarizes the reward formulations for each arm-motion condition, showing how shared locomotion objectives and arm–leg terms enforce normal swing, implement anti-normal (in-phase) by negating the hip-angle signals θ_{Lh} and θ_{Rh} before evaluating Eq (3.7), so the same indicator remains positive for the intended mode, and make arm-term contributions negligible under fixed arms. Fairness of comparison across arm-swing conditions. Accordingly, reward/Q0 comparisons are reported only for normal versus anti-normal, while fixed arms are excluded from reward/Q0 cross-condition comparisons due to the omission of arm-related shaping terms.

Table 3 lists the exact reward definition used in each condition. The base locomotion objective is

identical across all settings, targeting forward progression, upright stability, and effort regularisation. The only condition-dependent part concerns the arm-related shaping, which is included for normal and anti-normal to specify the intended coordination mode, and omitted for fixed arms because the shoulders are locked and unactuated. For this reason, reward-based quantities, such as total return and Q0, are interpreted within-condition, and cross-condition conclusions are drawn from reward-agnostic post-hoc steady-state gait and energy metrics computed identically for all policies. Specifically, we report V_y, $|V_x|$, dz, |X|, |P|, |τ|, W_{step}, W_{dist}, and CoT, which depend only on the simulated motion and actuation.

3.5. SAC hyperparameters

Controlling a humanoid robot is challenging owing to its high-dimensional action space and complex dynamics. To obtain more stable and efficient learning, some of the agent's hyperparameter values should be tuned to the operating context. The hyperparameter values reported in Table 4 are those used in non-default settings.

Table 4. Agent hyperparameter values.

Hyperparameters	Value
Sample time (control frequency)	0.025
Batch size	256
Replay buffer length	1e+6
Target smooth factor (polyak for target network)	5e-3
Warm start steps (initial random steps before learning)	1024
Number of epochs per update	2
Actor learning rate	1e-4
Critic learning rate	5e-4
Gradient clipping (gradient threshold)	1
L2 regularization (weight decay)	1e-5

A short sampling period of 0.025 s provides fast feedback and markedly increases the control frequency [27]. Using a medium-sized batch stabilises gradient estimates and lowers the per-update computational cost, enabling more efficient learning [19,28]. A large replay buffer lets the agent draw on a more diverse state history, helping to avoid over-fitting to recent experiences [29]. The chosen target smoothing factor updates the target networks slightly faster; setting it too high can cause instability [19]. Warm start denotes the number of random interaction steps collected before any network updates. Choosing warm start steps larger than the mini-batch size ensures that learning starts only after the replay buffer contains at least one full batch of experience, which improves early training stability [30]. Setting per-episode epochs to 2 means that at each update phase the agent iterates over the sampled data twice rather than once. Lower learning rates were set for both actor and critic to temper updates and avoid policy oscillations from overly aggressive steps. With the chosen gradient threshold, if the gradient norm exceeds 1 it is rescaled to 1, preventing exploding gradients. L2 regularisation penalises large weights to mitigate over-fitting; combined with a large replay buffer and batch size, this keeps the risk of memorising a small subset of data low [31–33]. Each algorithm was trained with five independent random seeds to control for stochasticity [34]. The seed sets the initial

network weights, the mini-batch sampling order in experience replay, the exploration noise, and any stochastic environment restarts [35]. The training budget and hyperparameters were held constant across all runs.

Using Bayesian optimisation, we tuned three hyperparameters that most strongly affect agent learning performance (mini-batch size, actor learning rate, and critic learning rate). The algorithm explores different hyperparameter combinations and searches for the configuration that maximises the average episodic reward. The best values found within the search space are reported in Table 4. After a few random evaluations, a Gaussian process (GP) model is updated with the available observations to obtain the posterior distribution of the objective. From this posterior we define an acquisition function $\mathcal{A}(\lambda)$ that quantifies the expected improvement of each candidate; it is maximized to select the next evaluation point λ^{+}. Let $F(\lambda)$ denote the average episodic reward achieved with hyperparameters λ, and let the current best value be $m^{\star}$. Then, the expected improvement is obtained as in Eq (3.9) [36–38]:

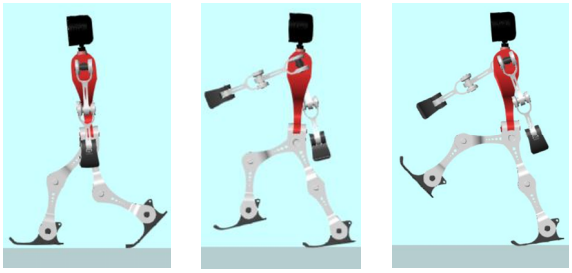
$$\mathcal{A}(\lambda) = EI(\lambda) = E_{\mathrm{G}}[\max(0,\, F(\lambda) - m^{\star})] \;\text{ where }\; m^{\star} = \max_{j \le t} F\left(\lambda_j\right) \tag{3.9}$$

Here, $E_{\mathrm{G}}[\cdot]$ denotes the expectation under the current GP model. Assume that the GP's posterior prediction at λ is $F(\lambda) \sim \mathcal{N}\left(\hat{\mu}(\lambda),\, \hat{\sigma}^2(\lambda)\right)$; in that case, $EI(\lambda)$ yields an expected gain based on both the probability that the predicted mean exceeds m^{*} and the magnitude by which it does so. This quantity can be computed analytically from the GP prediction and obtained for every candidate λ. At each iteration, the algorithm selects the point that maximises $EI(\lambda)$, runs the actual evaluation, adds the observation to the GP model to update the posterior distribution, and proceeds to the next iteration. The process continues until the iteration or time limit is reached, and the hyperparameter set that achieves the highest observed reward among the evaluations is deemed optimal.

In this study, learning curves and gait time-series are summarised across seeds using the seed median with interquartile range (IQR), where the central line denotes the median and the shaded band denotes the IQR (n = 5 seeds per condition). Between-condition differences are tested on seed-level summary values using the Kruskal–Wallis (KW) test. For KW, the effect size is reported as η_H^2. Steady-state metrics reported in the tables are computed per seed and then aggregated at the group level (reported as group mean and dispersion). For energy/effort interpretation, normalised measures such as $|P|/V_y$ and $|\tau|/V_y$ are used where appropriate [39,40].

4. Simulation

An integrated training and simulation environment was developed in MATLAB/Simulink using the reinforcement learning, deep learning, and Simscape multibody toolboxes. In addition, a URDF model of MATLAB's humanoid robot, as shown in Figure 2, was used. The robot's legs and arms are driven and stabilised by torque-controlled revolute joints (shoulder, hip, knee, and ankle), designed to mimic how humans use muscles in these two limbs while stepping. The shoulder joint has a mechanical axis that permits motion only in the sagittal plane, that is, rotation about the shoulder pitch axis (forward–backward swing) [41].



(a) (b) (c)

Figure 2. Gait configurations: (a) fixed–both arms are fully locked; hence no arm swing is allowed; (b) anti-normal–arms swing in-phase with the ipsilateral leg; (c) normal–human-like counter-phase arm–leg coordination. In (b) and (c), only the shoulder joint is actuated, while the remaining arm joints are locked; furthermore, the shoulder motion is constrained to the sagittal plane. All panels are shown at the same scale, viewed from the robot's left side, and at the same walking speed.

As illustrated in Figure 2, the robot's locomotion is modelled in 3D, yet the leg and arm motions are restricted to sagittal-plane kinematics for the gait comparisons. Three arm configurations are considered. In the fixed configuration (Figure 2(a)), the entire arm kinematic chain is immobilised by locking all arm joint axes, thereby eliminating arm swing. In the anti-normal configuration (Figure 2(b)) and the normal configuration (Figure 2(c)), the arm kinematic chain is simplified such that only the shoulder remains active, while distal arm joints are kept fixed. In these two conditions, arm motion is further constrained to a pure forward–backward shoulder rotation, and the difference between (b) and (c) is defined solely by the phase relationship between arm swing and leg motion: ipsilateral in-phase for anti-normal and contralateral anti-phase (counter-phase) for normal [11]. Table 5 summarizes the arm-related modelling and control settings used in the fixed, anti-normal, and normal configurations, including the active shoulder DoF(s), torque actuation, distal joint locking, motion-plane constraints, phase rules, and the corresponding action-space dimension (6 for fixed; 8 for anti-normal and normal).

During training, the model was enabled to interpret its interactions with the environment by continuously monitoring contact forces, the body's position and orientation in space (kinematic parameters), joint angles, forward-motion components, and the interaction of the arms, legs and torso.

Table 5. Definitions of the three gait configurations (normal, anti-normal, fixed): active shoulder DoF(s), shoulder torque actuation, locking of distal arm joints, arm-motion plane constraint, imposed arm–leg phase rule, and the resulting action-space dimension.

Condition	Shoulder DoFs (per arm)	Shoulder torque applied	Distal arm joints	Arm motion plane	Phase rule	Action space
Normal	1 (shoulder sagittal rotation only)	Yes	Locked	Sagittal only	Contralateral anti-phase	8 (6 leg + 2 shoulder)
Anti-normal	1 (shoulder sagittal rotation only)	Yes	Locked	Sagittal only	Ipsilateral in-phase	8 (6 leg + 2 shoulder)
Fixed	0 (all arm DoFs locked)	No	Locked	None	N/A	6 (only leg)

Training was conducted on a flat surface. First, data were obtained in simulation on the same flat

ground; thereafter, the performance of these trained agents was re-evaluated on an uneven surface to measure their responses to environmental changes. During training and simulation, the forward (y-axis), lateral (x-axis), and vertical (z-axis) directions were defined with respect to the robot's body coordinate frame.

ES–SAC training was carried out on a high-performance workstation (16-core AMD Ryzen 9 9950X3D CPU, NVIDIA GeForce RTX 5090 GPU, 64 GB RAM). Each condition was trained for 750 generations, with the episode length capped by the maximum step count. To reduce wall-clock time, we enabled parallel evaluation of individuals during training.

Table 6. Effects on average reward and gait quality score (Q0); values are median [IQR], n = 5.

Condition	Average reward (median [IQR] n = 5)	Q0 (median [IQR] n = 5)
Normal	8760 [8049–8960]	92.3 [82.5–100]
Anti-normal	8020 [7207–8220]	82.5 [78.4–86.3]
Fixed arms	1540 [1400–1680]	54.4 [51.6–61.0]

The ES parameter settings were as follows. The elite fraction was set to 50%. Accordingly, the best 12 individuals were designated as elites and carried forward to the next generation. The "returned policy" was defined as the highest-performing policy at the end of training and was stored as the final solution. A "weighted mixing" population update was employed; thus, in each subsequent generation, new policies were formed as weighted combinations of the elite parameters. The train epochs parameter denotes the number of SAC policy updates each individual performs per generation; in this study it was set to 50, ensuring that every individual had adequate opportunity to learn within a generation. The initial standard deviation was set to 0.25 to preserve sufficient diversity in parameter perturbations while preventing excessive oscillations during learning. The evaluations per individual parameter was set to 1, meaning each individual was assessed from a single simulation rollout. This choice shortened per-individual evaluation time and, in turn, substantially reduced the overall training time. The episode length was fixed across conditions and computed from the simulation horizon and sampling time. Looking at the training performance on flat terrain in Table 6, the anti-normal case stays in the same performance band as normal based on seed medians, yet it is ~8–9% lower in average reward. Therefore, reward-based quantities are compared only between normal and anti-normal; fixed arms are excluded from reward/Q0 cross-condition comparisons because the arm-related shaping terms are omitted.

In the gait statistics of Table 7, although anti-normal increases the forward speed (V_y) compared with normal (1.6189 vs 1.5423 m/s, about +5.0%), it slightly worsens path keeping, with higher $|V_x|$ (0.1747 vs 0.1678 m/s) and higher |X| (0.2280 vs 0.2242 m). The mean vertical displacement dz is also slightly more negative in anti-normal (-0.0599 vs -0.0545 m). Accordingly, power per unit speed ($|P|/V_y$ denotes the absolute total joint power normalised by the mean forward speed over the same steady-state window, with units of J/m.) is lower under normal than anti-normal (545.91 vs 557.14), indicating better energy economy under normal. By contrast, anti-normal requires less torque per speed, with ($|\tau|/V_y$) lower by about 9.8% (189.61 vs 210.15). However, normal delivers a higher return and a lower $|P|/V_y$ at comparable speeds (1.54 vs 1.62 m/s). With fixed arms, V_y drops markedly to 1.0094 m/s, and total power and torque are substantially lower than the other groups (|P| = 539.07 W; $|\tau|$ = 197.76 N·m). After normalisation, $|P|/V_y$ and $|\tau|/V_y$ are of the same order as normal (534.03

vs 545.91, and 195.91 vs 210.15), Therefore, the primary trade-off in the fixed-arms condition is reduced speed and degraded gait quality, and the small differences in normalised cost should not be interpreted as an efficiency gain.

Table 7. Gait metrics in the steady-state window (on flat terrain).

Condition	V_y	$\lvert V_x \rvert$	dz	|X|	|P|	$\lvert P \rvert / V_y$	|τ|	$\lvert \tau \rvert / V_y$
Normal	1.5423	0.1678	-0.0545	0.2242	841.93	545.91	324.11	210.15
Anti-normal	1.6189	0.1747	-0.0599	0.2280	901.97	557.14	306.97	189.61
Fixed arms	1.0094	0.1410	-0.0586	0.1810	539.07	534.03	197.76	195.91

**Note*: Coordinates: forward y, lateral x, vertical z. V_y: forward speed (m/s); $|V_x|$: mean absolute lateral speed (m/s); dz: mean vertical displacement (m); |X|: mean absolute lateral position (m); |P|: mean absolute total joint power (W); $|P|/V_y$: power per speed (J/m); |τ|: mean absolute total joint torque (N·m); $|\tau|/V_y$: torque per speed (N·m·s/m).

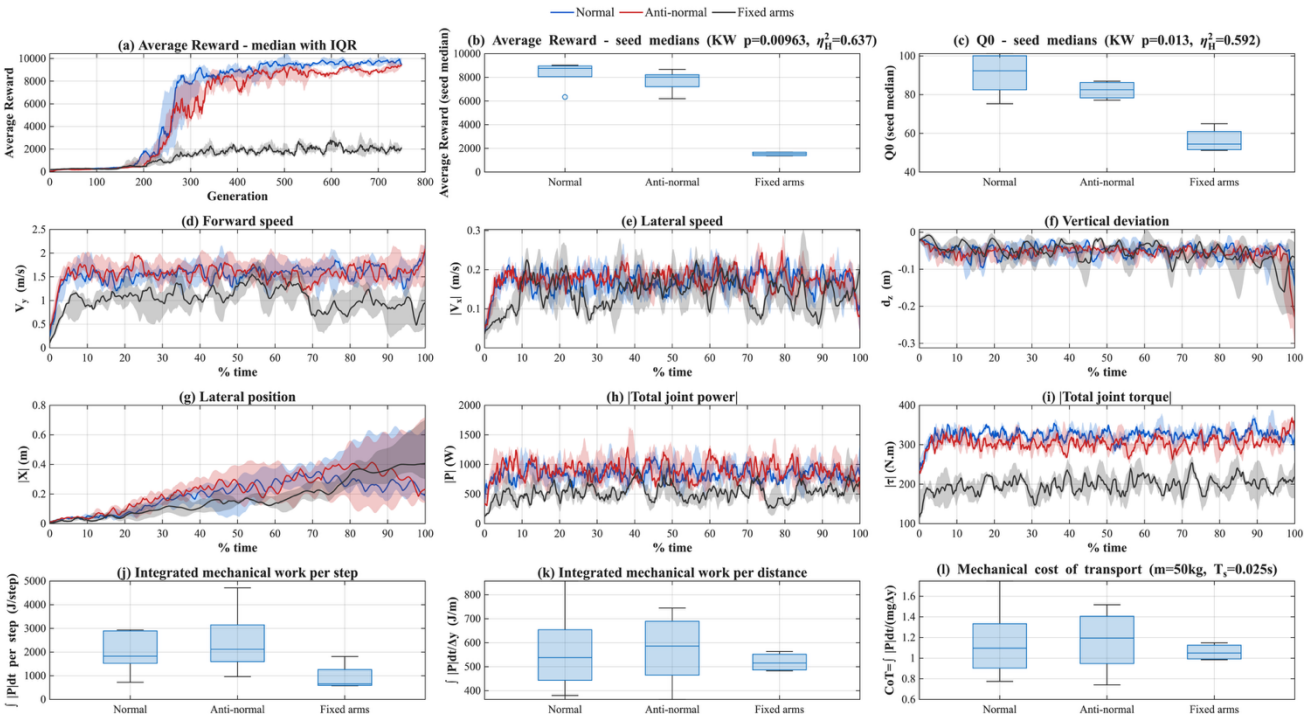


Figure 3. Learning, steady-state gait, and energy economy across arm-movement configurations on flat terrain; (a) Average reward across generations (seed median with IQR); (b-c) seed-median Average reward and Q0 (KW p and η_H^2 as shown); (d-i) V_y, $|V_x|$, d_z, |X|, |P|, and |τ| over the gait cycle (seed median with IQR shading); (j) integrated mechanical work per step, $\int |P|\ dt$ (boxplots across seeds); (k) integrated mechanical work per distance, $\int |P|\, dt\ /\ \Delta y$ (boxplots across seeds); (l) mechanical cost of transport, $CoT = \int |P|\, dt\ /(m\ g\ \Delta y)$ (boxplots across seeds; m and T_s indicated).

Figure 3(a) indicates a clear convergence trend in training: after an initial growth phase, the seed-median average reward enters a plateau where further improvements become marginal. In this steady region, normal saturates earlier and remains consistently near the upper reward band, while anti-normal

reaches a comparable level but shows a slightly weaker upper envelope and a less persistent plateau than normal. This stabilization of average reward is taken as evidence that the policy has largely converged under the given training budget and hyperparameters. Moreover, the IQR band around the seed median remains relatively stable in the plateau region, indicating reduced across-seed variability and reinforcing the convergence interpretation. Figure 3(b) reports seed medians of 8760 (IQR 8049–8960) for normal and 8020 (7207–8220) for anti-normal. The KW test yields p = 0.0096 and η_H^2 = 0.637, indicating a significant difference. Thus, normal is superior in training reward to anti-normal. In Figure 3(c) the difference is again significant (p = 0.013, η_H^2 = 0.592). Looking at Q0, normal also leads in initial/early quality: 92.3 (82.5–100) for normal versus 82.5 (78.4–86.3) for anti-normal. These reward-based comparisons are restricted to normal versus anti-normal; accordingly, fixed arms are not compared on reward/Q0. In Figure 3(d) the mean forward speed is 1.6189 m/s for anti-normal and 1.5423 m/s for normal, so anti-normal is ≈ 5.0% faster. Figure 3(e) gives lateral speed; the means are close (normal 0.1678, anti-normal 0.1747), with anti-normal slightly higher, which makes normal preferable. Figure 3(f) shows vertical oscillation; values are very similar: -0.0545 for normal and -0.0599 for anti-normal. For lateral deviation (Figure 3(g)), normal is slightly better: 0.2242 versus 0.2280 for anti-normal, that is., ≈ +1.7% relative to normal, indicating marginally straighter path keeping. Figure 3(h) presents the total joint power, while Figure 3(i) reports the total joint torque. Mean total joint power is higher for anti-normal (901.97) than normal (841.93), that is, ≈ +7%. After normalisation by speed ($|P|/V_y$), anti-normal is 557.14 versus 545.91 for normal, about ≈ +2% more costly, so energy economy favours normal. Anti-normal's total $|\tau|$ is lower (306.97 vs 324.11 for normal); after speed normalisation the gap widens: $|\tau|/V_y$ is 189.61 vs 210.15 (≈ -9.8%), indicating better torque efficiency for anti-normal. From a torque-per-speed perspective, anti-normal shows a more favourable trade-off. The energy metrics shown in panels Figure 3(j)–(l) quantitatively reveal the trade-off between kinematic performance and energetic economy. Within the steady-state window, the time integral of the absolute joint power was used to compute the mechanical work per step, $W_{step} = \int |P|\, dt\ (J/step)$, the mechanical work per unit distance, $W_{dist} = \int |P|\, dt/\Delta y\ (J/m)$, and the mechanical cost of transport, $CoT = \int |P|\, dt/(m\ g\ \Delta y)$ is evaluated over the steady-state window and represents the mechanical cost of transport, normalised by body weight and forward distance, and is reported as a dimensionless quantity.

The resulting median values were $W_{\text{step}} \approx 1.83 \times 10^3 J/step$, $W_{\text{dist}} \approx 5.4 \times 10^2 J/m$, and CoT ≈ 1.1 for normal; $W_{\text{step}} \approx 2.12 \times 10^3 J/step$, $W_{\text{dist}} \approx 5.86 \times 10^2 J/m$, and CoT ≈ 1.2 for anti-Normal; and $W_{\text{step}} \approx 6.56 \times 10^2 J/step$, $W_{\text{dist}} \approx 5.15 \times 10^2 J/m$, and CoT ≈ 1.1 for fixed arms (Figure 3(j)–(l)). Because fixed operates at a substantially lower V_y, its work and CoT proxies are reported for completeness and should not be interpreted as a speed-independent economy advantage. The box plots quantify the energetic trade-off between the two swinging strategies. Normal and anti-normal operate in a comparable locomotion regime, and anti-normal tends to show higher mechanical-work proxies (W_{step}, W_{dist}, and CoT), consistent with its higher $|P|/V_y$, despite being faster and requiring less torque per speed. Hence, energy-based comparisons are most informative between normal and anti-normal, where the locomotion regimes are closer and the trade-offs can be interpreted more fairly. Fixed arms are shown for completeness, but its lower work/CoT proxies should not be interpreted as a like-for-like efficiency improvement because it corresponds to a distinct, lower-speed locomotion regime.

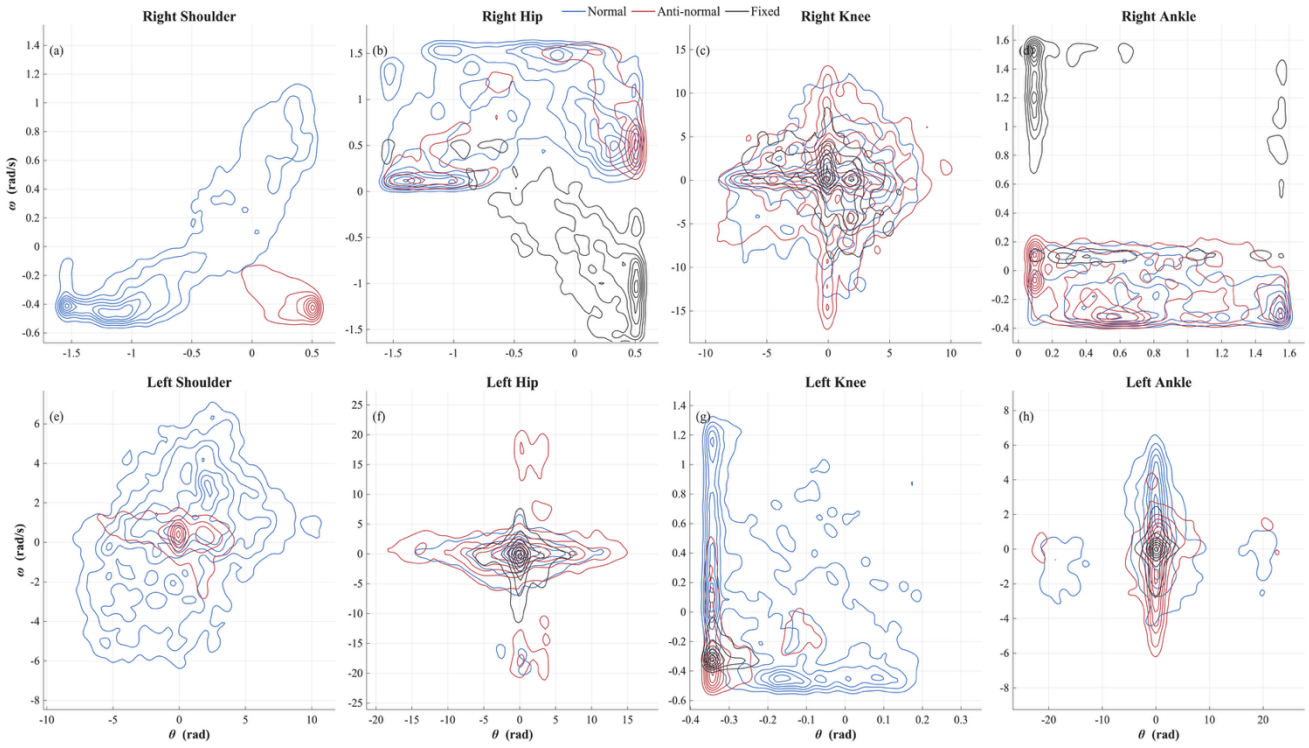


Figure 4. Event-aligned limit-cycle phase portraits (density contours) of joint angle versus joint angular velocity for the three arm-swing configurations. Shoulders: normal and anti-normal; lower limbs (hip, knee, ankle): fixed, normal, and anti-normal.

Figure 4 summarizes the steady-state gait dynamics using phase portraits of joint angle versus joint angular velocity. The portraits are constructed by extracting samples around detected foot-contact events and aligning them across steps and seeds, so that repeatedly visited closed loops represent a periodic limit cycle while the contour spread reflects step-to-step variability. For the shoulders, normal and anti-normal exhibit distinct limit-cycle organizations. Normal yields a more compact and repeatable shoulder cycle, whereas anti-normal shows a broader occupancy and locally increased dispersion, indicating less consistent upper-body coordination across steps. This is consistent with the imposed coordination patterns: normal promotes contralateral, opposite-phase arm motion, while anti-normal enforces an in-phase pattern that can couple more strongly with trunk and stance dynamics. For the lower-limb joints, all configurations maintain periodic locomotion, but the loop geometry and dispersion differ across conditions. Fixed arms generally produce wider and more scattered hip, knee, and ankle portraits, suggesting increased variability in leg coordination when the upper body is constrained. In contrast, normal tends to concentrate probability mass into a more coherent loop, indicating improved regularity of the limit cycle. Anti-normal remains periodic but often presents a more complex loop structure and a larger spread in parts of the cycle, consistent with additional coupling demands induced by the in-phase upper-body motion.

The uneven terrain (Figure 5) has the same dimensions as the flat surface (50 m in length and 3 m in width) but, unlike the flat case, it is generated as a stochastic rough track rather than a single hand-crafted profile. Specifically, the ground height is constructed from a random, spatially correlated surface with centimetre-scale rounded depressions and humps; the correlation length is set to approximately 1 m so that slope changes persist over several steps instead of appearing as isolated

spikes. Importantly, this randomness is distributed along the full 50 m track: the perturbations are not confined to a local patch but are spread throughout the entire walking corridor, creating a continuous sequence of uneven contacts. The terrain generator is parameterised (e.g., by RMS height, correlation length, and depression/hump density) and controlled via a random seed, allowing multiple terrain realisations with the same statistical properties to be produced for robustness testing. This ground challenges the robot in four ways: risk of foot scuffing/mistimed contact during swing; loss of moment balance in stance due to partial foot contact and variable normal forces; lateral drift under asymmetric slopes with frequent speed corrections and increased power/torque demand; and, finally, a shift in the parameters learned on flat ground. Accordingly, the contribution of arm-swing patterns (normal vs anti-normal) to path keeping and balance becomes clearly observable on this terrain.

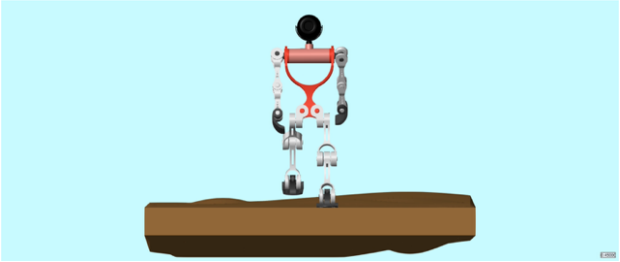


Figure 5. Humanoid walking on uneven terrain.

Table 8. Gait metrics in the steady-state window (on uneven terrain).

Condition	V_y	$\lvert V_x \rvert$	dz	|X|	|P|	$\lvert P \rvert / V_y$	|τ|	$\lvert \tau \rvert / V_y$
Normal	1.5929 [0.68]	0.1662 [0.03]	0.0089 [0.01]	0.4517 [0.36]	768.33 [187]	482.35 [185]	324.28 [13]	203.58 [82]
Anti-normal	1.662 [0.6]	0.1615 [0.01]	0.0002 [0.02]	0.2843 [0.17]	920.95 [442]	554.13 [164]	314.39 [28]	189.17 [61]
Fixed arms	1.0644 [0.23]	0.1165 [0.03]	-0.0477 [0.02]	0.3294 [0.06]	522.48 [50]	490.88 [146]	198.88 [15]	186.85 [79]

**Note:* Coordinates: forward y, lateral x, vertical z. V_y: forward speed (m/s); $|V_x|$: mean absolute lateral speed (m/s); dz: mean vertical displacement (m); |X|: mean absolute lateral position (m); |P|: mean absolute total joint power (W); $|P|/V_y$: power per speed (J/m); |τ|: mean absolute total joint torque (N·m); $|\tau|/V_y$: torque per speed (N·m·s/m). Values are reported as median [∓IQR]; each policy seed is evaluated on five terrain seeds.

On uneven terrain, the evaluation protocol uses five independent terrain realizations (terrain seeds), and for each terrain seed, five independently trained policy seeds are tested for each condition (normal, anti-normal, and fixed). The same set of terrain seeds is used across all conditions to ensure a fair comparison. For each terrain seed, we first aggregate the five policy-seed trials using the median and then report the distribution across the five terrain seeds using the median and IQR. When the same policies are compared on flat terrain (Table 7) versus uneven terrain (Table 8), two clear tendencies emerge. First, the anti-normal policy keeps its advantage on the speed-torque axis; second, on uneven ground the anti-normal policy strengthens path keeping, whereas normal remains relatively stronger in power-per-speed. Anti-normal is faster on both terrains: it is about +5.0% over normal on the flat

(1.6189 vs 1.5423 m/s), and the gap remains at about +4.3% on the uneven ground (1.662 vs 1.5929 m/s). In terms of lateral accumulation (|X|), normal is slightly better on the flat (0.2242 m vs 0.2280 m for anti-normal, ≈ +1.7% higher |X| in anti-normal), but on uneven ground anti-normal is better (0.2843 m vs 0.4517 m, ≈ 37% lower |X| than normal). For instantaneous lateral speed ($|V_x|$), anti-normal shows higher lateral oscillation on the flat (0.1747 vs 0.1678 m/s), whereas on uneven ground it is slightly lower (0.1615 vs 0.1662 m/s). The power-per-speed cost ($|P|/V_y$) is lower for normal on both terrains, about +2.1% on the flat (557.14 vs 545.91) and about +14.9% on the uneven (554.13 vs 482.35). The torque-per-speed ($|\tau|/V_y$) advantage remains with anti-normal: about -9.8% on the flat (189.61 vs 210.15) and about -7.1% on the uneven (189.17 vs 203.58). Vertical displacement (dz) remains small in magnitude: on the flat anti-normal is slightly more negative (-0.0599 vs -0.0545 m), whereas on uneven terrain both are close to zero, with normal slightly higher (0.0089 vs 0.0002 m).

Collins' human walking experiments ([1]) indicate that natural (normal) arm swing makes gait more economical, whereas restricting the arms (fixed) and anti-normal (ipsilateral, in-phase) arm swing generally tend to increase the energetic cost. In the present study, although metabolic energy is not measured directly, the simulation results reveal a consistent separation when assessed through mechanical power/torque-based proxies (normal remains closer to a lower-cost regime, anti-normal shows a higher-cost tendency, and fixed exhibits a distinctly different regime). In addition, the learning curves suggest that the arm-swing configuration directly affects learning efficiency: the normal condition reaches a high-reward band at earlier generations and exhibits faster, more stable convergence, whereas anti-normal attains a comparable level but with a lower median/upper bound and a later plateau, indicating a weaker learning profile.

5. Conclusions

A reduced-DoF spatial (3D) humanoid walker with predominantly sagittal-plane motion was trained with ES–SAC under identical hyperparameters and conditions, and three arm strategies (normal, anti-normal, and fixed arms) were compared on both flat and uneven terrain (five independent random seeds per condition). Between the two swinging strategies, swing was faster on both terrains, yet its torque-per-speed was lower (better joint-effort efficiency). By contrast, on uneven terrain the anti-normal swing kept path tracking more stable (lower |X|), whereas the normal swing maintained a lower power-per-speed ($|P|/V_Y$), so the trade-off separates into better tracking for anti-normal and better energy economy for normal when the ground is irregular. Collins et al. [1] showed that fixing the arms raises metabolic cost by ≈ 12%, in-phase arm swing can raise it by up to 26%, and fixing the arms increases the vertical reaction moment by ≈ 63%, whereas normal swing requires little shoulder torque. Based on the proxy metrics, fixed exhibits a distinct regime primarily driven by the reduced speed. The numerical gaps differ because our "energy" is mechanical (sum of absolute joint power, $|\tau \cdot \omega|$), not metabolic; human studies report net metabolic power. Our simulator does not model basal metabolism or actuator/electronics losses, so absolute percentages are not interchangeable (only indicative of trends). In short, ES–SAC recovers the human insight that arm swing is beneficial: if speed and torque efficiency are the priority, then anti-normal is superior; when the ground is uneven, anti-normal swing yields better lateral tracking, while normal swing yields lower power-per-speed ($|P|/V_y$). Overall, these conclusions accord with studies on biped gait stability [42] and data-driven/robust control in complex dynamical systems [43], and they sit alongside recent work in robotics and human-robot studies [44,45]. This work is evaluated in simulation. Transfer to hardware can be affected by contact and friction

mismatch, actuator saturation, and sensor noise and delays. As future work, we plan hardware deployment using the Simulink-to-embedded workflow and will report real-world results separately.

Use of AI tools declaration

The authors declare that no generative artificial intelligence (AI) tools were used to write, analyze, or create any content (text, figures, or code) for this manuscript.

Conflict of interest

No potential competing interest was reported by the authors.

References

1. S. H. Collins, P. G. Adamczyk, A. D. Kuo, Dynamic arm swinging in human walking, *Proc. R. Soc. B Biol. Sci.*, **276** (2009), 3679–3688. https://doi.org/10.1098/rspb.2009.0664
2. N. Itahashi, H. Itoh, H. Fukumoto, H. Wakuya, Reinforcement learning of bipedal walking using a simple reference motion, *Appl. Sci.*, **14** (2024), 1803. https://doi.org/10.3390/app14051803
3. F. Wu, Z. Gu, H. Wu, A. Wu, Y. Zhao, Infer and adapt: Bipedal locomotion reward learning from demonstrations via inverse reinforcement learning, in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, (2024), 16243–16250. https://doi.org/10.48550/arXiv.2309.16074
4. Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, K. Sreenath, Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control, preprint, arXiv:2401.16889. https://doi.org/10.48550/arXiv.2401.16889
5. R. P. Singh, Z. Xie, P. Gergondet, F. Kanehiro, Learning bipedal walking for humanoids with current feedback, *IEEE Access*, **11** (2023), 82013–82023. https://doi.org/10.1109/ACCESS.2023.3301175
6. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, et al., Continuous control with deep reinforcement learning, preprint, arXiv:1509.02971. https://doi.org/10.48550/arXiv.1509.02971
7. S. Fujimoto, H. van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in *International Conference on Machine Learning*, PMLR, (2018), 1587–1596. https://doi.org/10.48550/arXiv.1802.09477
8. N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, et al., Emergence of locomotion behaviours in rich environments, preprint, arXiv:1707.02286. https://doi.org/10.48550/arXiv.1707.02286
9. X. Zhang, X. Wang, L. Zhang, G. Guo, X. Shen, W. Zhang, Achieving stable high-speed locomotion for humanoid robots with deep reinforcement learning, preprint, arXiv:2409.16611. https://doi.org/10.48550/arXiv.2409.16611
10. X. Wang, W. Guo, S. Yin, S. Zhang, F. Zha, M. Li, et al., Walking control of humanoid robots based on improved footstep planner and whole-body coordination controller, *Front. Neurorob.*, **19** (2025), 1538979. https://doi.org/10.3389/fnbot.2025.1538979

11. H. Herr, M. Popovic, Angular momentum in human walking, *J. Exp. Biol.*, **211** (2008), 467–481. https://doi.org/10.1242/jeb.008573
12. B. R. Umberger, Effects of suppressing arm swing on kinematics, kinetics, and energetics of human walking, *J. Biomech.*, **41** (2008), 2575–2580. https://doi.org/10.1016/j.jbiomech.2008.05.024
13. A. Pourchot, O. Sigaud, CEM-RL: Combining evolutionary and gradient-based methods for policy search, preprint, arXiv:1810.01222. https://doi.org/10.48550/arXiv.1810.01222
14. S. Khadka, K. Tumer, Evolution-guided policy gradient in reinforcement learning, *Adv. Neural Inf. Process. Syst.*, **31** (2018). https://doi.org/10.48550/arXiv.1805.07917
15. A. Roaas, G. B. J. Andersson, Normal range of motion of the hip, knee and ankle joints in male subjects, 30–40 years of age, *Acta Orthop. Scand.*, **53** (1982), 205–208. https://doi.org/10.3109/17453678208992202
16. J. M. Soucie, C. Wang, A. Forsyth, A. Funk, S. Denny, M. Roach, et al., Hemophilia treatment center network, range of motion measurements: reference values and a database for comparison studies, *Haemophilia*, **17** (2011), 500–507. https://doi.org/10.1111/j.1365-2516.2010.02399.x
17. T. Salimans, J. Ho, X. Chen, S. Sidor, I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning, preprint, arXiv:1703.03864. https://doi.org/10.48550/arXiv.1703.03864
18. N. Hansen, The CMA evolution strategy: A tutorial, preprint, arXiv:1604.00772. https://doi.org/10.48550/arXiv.1604.00772
19. T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor, in *International Conference on Machine Learning*, (2018), 1861–1870. https://doi.org/10.48550/arXiv.1801.01290
20. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Venesset, M. G. Bellemare, et al., Human-level control through deep reinforcement learning, *Nature*, **518** (2015), 529–533. https://doi.org/10.1038/nature14236
21. K. Suri, X. Q. Shi, K. N. Plataniotis, Y. A. Lawryshyn, Maximum mutation reinforcement learning for scalable control, preprint, arXiv:2007.13690. https://doi.org/10.48550/arXiv.2007.13690
22. K. Lee, B. U. Lee, U. Shin, I. S. Kweon, An efficient asynchronous method for integrating evolutionary and gradient-based policy search, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 10124–10135. https://doi.org/10.48550/arXiv.2012.05417
23. M. Calì, A. Sinigaglia, N. Turcato, R. Carli, G. A. Susto, Finetuning deep reinforcement learning policies with evolutionary strategies for control of underactuated robots, *IFAC-PapersOnLine*, **59** (2025), 31–36. https://doi.org/10.1016/j.ifacol.2025.12.006
24. K. Suri, X. Shi, K. N. Plataniotis, Y. A. Lawryshyn, Evolve to control: evolution-based soft actor-critic for scalable reinforcement learning, preprint, arXiv:2007.13690. https://doi.org/10.48550/arXiv.2007.13690v1
25. X. B. Peng, G. Berseth, K. Yin, M. V. De Panne, DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning, *ACM Trans. Graphics*, **36** (2017), 1–13. https://doi.org/10.1145/3072959.3073602
26. I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, K. Sreenath, Real-world humanoid locomotion with reinforcement learning, *Sci. Rob.*, **9** (2024), eadi9579. https://doi.org/10.1126/scirobotics.adi9579

27. A. A. Issa, A. A. Aldair, Learning the quadruped robot by reinforcement learning (RL), *Iraqi J. Electr. Electron. Eng.*, **18** (2022), 117–126. https://doi.org/10.37917/ijeee.18.2.15
28. K. Hong, Y. Li, A. Tewari, A primal-dual-critic algorithm for offline constrained reinforcement learning, in *International Conference on Artificial Intelligence and Statistics*, (2024), 280–288. https://doi.org/10.48550/arXiv.2306.07818
29. R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge: MIT press, 1998.
30. A. Louette, G. Lambrechts, D. Ernst, E. Pirard, G. Dislaire, Reinforcement learning to improve delta robot throws for sorting scrap metal, preprint, arXiv:2406.13453. https://doi.org/10.48550/arXiv.2406.13453
31. D. Masters, C. Luschi, Revisiting small batch training for deep neural networks, preprint, arXiv:1804.07612. https://doi.org/10.48550/arXiv.1804.07612
32. D. Tarasov, A. Surina, C. Gulcehre, The role of deep learning regularizations on actors in offline RL, preprint, arXiv:2409.07606. https://doi.org/10.48550/arXiv.2409.07606
33. Z. Liu, X. Li, B. Kang, T. Darrell, Regularization matters in policy optimization: an empirical study on continuous control, in *International Conference on Learning Representations*, 2021. https://doi.org/10.48550/arXiv.2102.03050
34. P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, D. Meger, Deep reinforcement learning that matters, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **32** (2018). https://doi.org/10.1609/aaai.v32i1.11694
35. R. Islam, P. Henderson, M. Gomrokchi, D. Precup, Reproducibility of benchmarked deep reinforcement learning tasks for continuous control, preprint, arXiv:1708.04133. https://doi.org/10.48550/arXiv.1708.04133
36. B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. de Freitas, Taking the human out of the loop: A review of Bayesian optimization, *Proc. IEEE*, **104** (2016), 148–175. https://doi.org/10.1109/JPROC.2015.2494218
37. J. Snoek, H. Larochelle, R. P. Adams, Practical bayesian optimization of machine learning algorithms, *Adv. Neural Inf. Process. Syst.*, **25** (2012), 2951–2959. https://doi.org/10.48550/arXiv.1206.2944
38. M. A. Gelbart, J. Snoek, R. P. Adams, Bayesian optimization with unknown constraints, preprint, arXiv:1403.5607. https://doi.org/10.48550/arXiv.1403.5607
39. R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, M. G. Bellemare, Deep reinforcement learning at the edge of the statistical precipice, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 29304–29320. https://doi.org/10.48550/arXiv.2108.13264
40. H. B. Mann, D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, *Ann. Math. Stat.*, **18** (1947), 50–60. https://doi.org/10.1214/aoms/1177730491
41. Q. Wang, L. D. Baets, A. Timmermans, W. Chen, L. Giacolini, T. Matheve, et al., Motor control training for the shoulder with smart garments, *Sensors*, **17** (2017), 1687. https://doi.org/10.3390/s17071687
42. J. Chen, A. Tang, G. Zhou, L. Lin, G. Jiang, Walking dynamics for an ascending stair biped robot with telescopic legs and impulse thrust, *Electron. Res. Arch.*, **30** (2022), 4108–4135. https://doi.org/10.3934/era.2022208

43. Y. Chen, H. Zhao, M. Ogura, H. Yu, L. Peng, Data-driven event-triggered fixed-time load frequency control for multi-area power systems with input delays, *IEEE Trans. Circuits Syst. I Regul. Pap.*, **72** (2025), 8492–8504. https://doi.org/10.1109/TCSI.2025.3580122
44. Y. Dong, X. Zhou, Advancements in AI-driven multilingual comprehension for social robot interactions: An extensive review, *Electron. Res. Arch.*, **31** (2023), 6600–6633. https://doi.org/10.3934/era.2023334
45. Y. Lei, Z. Su, C. Cheng, Virtual reality in human-robot interaction: challenges and benefits, *Electron. Res. Arch.*, **31** (2023), 2374–2408. https://doi.org/10.3934/era.2023121

AIMS Press