Task-Aware Morphology Optimization of Planar Manipulators via Reinforcement Learning

Arvind Kumar Mishra * **Sohom Chakrabarty** *

* *Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee 247667,India*
(arvind_m@ee.iitr.ac.in; sohom@ee.iitr.ac.in)

Abstract: In this work,Yoshikawa's manipulability index is adopted as the basis for investigating reinforcement learning (RL) as a framework for morphology optimization in planar robotic manipulators. A 2R manipulator tracking a circular end-effector path was first examined, since this setting admits a closed-form analytical optimum that both links are equal in length and the second joint is orthogonal to the first. This case was used to validate that the optimum could be rediscovered from the RL framework with reward feedback alone, without access to the closed-form manipulability expression or the Jacobian model. Three RL algorithms—Soft Actor–Critic (SAC), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO)—were compared against grid search and black-box optimizers, with the morphology represented by a single action parameter ϕ that maps to (L_1, L_2) under the circular constraint. All methods converged to the analytical solution, showing that the optimum can be recovered numerically even when the analytical model is not supplied.
Most morphology design problems do not admit closed-form solutions, and heuristic or grid search becomes increasingly expensive due to dimensionality and discretization effects. RL was therefore considered as a scalable alternative. The RL formulation and training setup used in the circular case were retained for elliptical and rectangular paths, while the action space were expanded from a single parameter ϕ to the full morphology vector (L_1, L_2, θ_2). In these non-analytical tasks, RL continued to converge reliably, whereas extending grid or black-box search to these cases would require substantially larger evaluation budgets due to the higher dimensionality of the morphology space.These results indicate that RL is suitable both for recovering known analytical optima and for morphology optimization in settings lacking closed-form solutions.

Keywords: reinforcement learning, morphology optimization, manipulability,planar manipulator, SAC, DDPG, PPO, bandit formulation.

1. INTRODUCTION

The morphology of a manipulator—defined by its link lengths and joint configuration—plays a central role in determining its dexterity, energy efficiency, and overall task performance (Craig (2005); Siciliano et al. (2010); Yoshikawa (1985)). Yoshikawa's manipulability index provides a classical measure of kinematic dexterity through the determinant of the Jacobian matrix (Yoshikawa (1985)). For a planar 2R manipulator, the measure of Yoshikawa gives a closed-form maximum corresponding to equal link lengths and an orthogonal joint configuration (Yoshikawa (1985); Craig (2005)). Specifically, for a circular path with the manipulator base at the centre, the analytical optimum occurs when

$$L_1 = L_2 = \frac{R}{\sqrt{2}}, \quad \theta_2 = 90^\circ,$$

which simultaneously maximizes manipulability and satisfies the geometric constraint $x^2 + y^2 = R^2$ (Yoshikawa (1985)). However, for manipulators with higher degrees of freedom (DOF) or for arbitrary task trajectories, the Jacobian becomes highly nonlinear and coupled, making analytical optimization infeasible (Craig (2005)). This motivates the use of numerical and data-driven learning techniques for morphology optimization.

Traditional optimization methods such as grid search and heuristic algorithms—including particle swarm optimization (PSO) (Kennedy and Eberhart (1995)), Bayesian optimization (BO) (Mockus et al. (1978); Jones et al. (1998)), and covariance matrix adaptation evolution strategy (CMA-ES) (Hansen (2006))—have been widely employed to explore robot design spaces. Although these general-purpose algorithms can identify near-optimal configurations, they often experience discretization errors, slow convergence, and poor scalability with increasing dimensionality (Hansen (2006); Jones et al. (1998)). Furthermore, they typically lack mechanisms to handle actuator-level constraints such as joint velocity or torque limits, restricting their utility in dynamic or physically realistic scenarios (Zhang and Song (2008); De Luca and Siciliano (1992); Gros and Zanon (2020)).

Recent works have extended these heuristic techniques directly to manipulator and robot morphology optimization.

For example, Farooq et al. (2021) used PSO to optimize the structural parameters of a Tricept parallel manipulator for enhanced dexterity, while Shahbazi et al. (2024) applied PSO for dynamic parameter design of Stewart manipulators. Similarly, CMA-ES has been successfully applied in soft-robot morphology design (Medvet et al. (2021)) and reviewed as a promising method in recent soft-robot optimisation surveys (Stroppa et al. (2024)). These methods, however, do not adapt well to time-varying constraints, such as changing workspace geometry, payload, or joint limits, since a full re-optimization is required whenever the conditions change.

Reinforcement Learning (RL) provides a flexible, model-free alternative capable of optimizing morphology in continuous action spaces under complex reward definitions (Sutton and Barto (2018); Haarnoja et al. (2018); Lillicrap et al. (2016); Schulman et al. (2017)). In this work, three continuous-control Deep RL algorithms—Soft Actor–Critic (SAC), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO)—are utilized. The morphology variables $[L_1, L_2, \theta_2]$ are represented as continuous actions within a single-state bandit framework, allowing the agent to explore and learn the optimal geometry through reward feedback. For the circular task, RL is benchmarked against the analytical optimum (Yoshikawa (1985)) and numerical baselines to validate its applicability. Once validated, the same framework is extended to non-circular trajectories such as elliptical and rectangular paths, where analytical solutions are unavailable.

The main contributions of this work are as follows:

- Verification of Yoshikawa's manipulability optimum using RL and heuristic methods on the circular path.
- Demonstration that RL can serve as a reliable reference for morphology optimization when analytical solutions do not exist.
- Analysis of the computational and scalability limitations of grid and heuristic algorithms (Jones et al. (1998); Hansen (2006)).
- Illustration of how the RL framework can be extended to handle dynamic constraints and higher-DOF manipulators.

The remainder of this paper is organized as follows. Section 2 reviews related work on manipulability and morphology optimization. Section 3 presents the analytical and heuristic formulations. Section 4 describes the RL framework and reward design. Section 5 reports comparative results for circular, elliptical, and rectangular tasks. Section 6 discusses and future scope, followed by conclusions in Section 7.

2. RELATED WORK

2.1 Manipulability and Morphology Optimization

The concept of manipulability, introduced by Yoshikawa (1985), is still widely used in evaluating the dexterity of robotic manipulators. It relates the Jacobian determinant to the isotropy of velocity transmission in Cartesian space. Subsequent works and textbooks, including Craig (2005) and Siciliano et al. (2010), formalized the relationship between manipulator morphology, joint configuration, and task-space performance. For planar 2R manipulators, analytical optimization reveals that equal link lengths and orthogonal joint configurations maximize manipulability when the circular path is centered at the manipulator base, providing a useful benchmark for validating computational methods.

2.2 Heuristic and Evolutionary Optimization

To overcome the limitations of analytical formulations, many studies have adopted numerical and heuristic optimization methods. Classical global optimizers such as particle swarm optimization (PSO) (Kennedy and Eberhart (1995)), Bayesian optimization (BO) (Mockus et al. (1978); Jones et al. (1998)), and covariance matrix adaptation evolution strategy (CMA-ES) (Hansen (2006)) have been widely applied to robot design and parameter tuning. Although these general-purpose techniques can identify near-optimal configurations, they typically incur high computational cost, slow convergence, and poor scalability with increasing design dimensionality (Hansen (2006); Jones et al. (1998)). Furthermore, they often neglect actuator-level constraints such as joint velocity and torque limits, limiting their applicability to dynamic systems (Zhang and Song (2008); De Luca and Siciliano (1992); Gros and Zanon (2020)). Recent works have extended these methods to manipulator morphology, such as PSO-based optimization of parallel and Stewart platforms (Shahbazi et al. (2024)) and CMA-ES-driven design of soft robotic structures ((Medvet et al., 2021; Stroppa et al., 2024)). While these studies demonstrate that heuristic algorithms can adapt to manipulator design, their reliance on dense sampling makes them computationally expensive and unsuitable for real-time or adaptive morphology optimization.The specific configurations and hyperparameters used for PSO, BO, and CMA-ES in this study are summarized in Table B.1.

2.3 Learning-Based and Reinforcement Learning Approaches

With the rise of deep reinforcement learning (DRL), several studies have explored learning-based co-design, where morphology and control are optimized jointly. RL provides a model-free formulation capable of handling continuous actions and complex rewards (Sutton and Barto (2018)). Algorithms such as SAC (Haarnoja et al. (2018)), DDPG (Lillicrap et al. (2016)), and PPO (Schulman et al. (2017)) enable continuous optimization without explicit gradient derivations of the kinematic or dynamic models. Learning-based co-design frameworks have been applied in areas such as tool design, legged locomotion, and soft-robot morphology evolution; these studies show reinforcement learning can scale effectively to complex design problems and adapt better to changing conditions than traditional heuristic methods (Haarnoja et al. (2018); Lillicrap et al. (2016); Schulman et al. (2017)).

2.4 Motivation and Scope

Despite these advances, a unified comparison between analytical, heuristic, and reinforcement learning approaches

for morphology optimization remains limited. In particular, the classical Yoshikawa benchmark has rarely been revisited under a learning-based framework. This study bridges that gap by verifying Yoshikawa's analytical result using both numerical and RL methods on a circular task, then extending the framework to non-circular trajectories where closed-form solutions are unavailable.

3. ANALYTICAL AND HEURISTIC FORMULATION

3.1 Analytical Benchmark: Planar 2R Manipulator

The planar two–revolute (2R) manipulator in Fig. 1 is used as the analytical benchmark for morphology optimization. Its end–effector Cartesian coordinates are given by the standard forward kinematics

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2), \tag{1a}$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2), \tag{1b}$$

where L_1, L_2 are the link lengths and θ_1, θ_2 are the joint angles.

The corresponding Jacobian is

$$J(\theta) = \begin{bmatrix} -L_1 \sin\theta_1 - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix}. \tag{2}$$

Yoshikawa's manipulability index (Yoshikawa (1985)) measures the isotropy of velocity transmission. For a planar 2R manipulator, the determinant of the Jacobian simplifies to

$$w(\theta_2) = \sqrt{\det(JJ^\top)} = L_1 L_2 |\sin\theta_2|. \tag{3}$$

Closed-form optimum for a centered circular task. Consider a circular end-effector trajectory of radius R centred at the base. Maximum manipulability is achieved when $|\sin\theta_2|$ attains its maximum value,

$$\theta_2 = 90^\circ \quad \Rightarrow \quad |\sin\theta_2| = 1.$$

Under this configuration, the reachability constraint reduces to

$$L_1^2 + L_2^2 = R^2, \tag{4}$$

which follows from the geometry of the manipulator at full extension.

Maximizing (3) subject to (4) yields the symmetric solution

$$L_1 = L_2 = \frac{R}{\sqrt{2}},$$

since the product $L_1 L_2$ is maximized when both links are equal. Substituting this result into (3) gives the peak manipulability

$$w_{\max} = R^2/2. \tag{5}$$

This equal-length configuration constitutes the analytical optimum for the centered circular path and is used as a *baseline morphology* throughout our experiments. For non-circular, off-centre, or constrained workspaces, no closed-form optimum exists, motivating the numerical and reinforcement-learning methods developed in the remainder of the paper.

3.2 Grid-Search Verification

To validate the analytical optimum derived in Section 3.1, a one-dimensional sweep over the morphology parameter ϕ

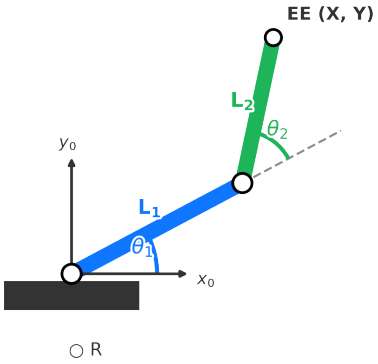

Fig. 1. Planar 2R manipulator schematic.

is performed instead of a full grid search over (L_1, L_2, θ_2). For a circular end-effector path of fixed radius R, the variables are related by the inverse-kinematics (IK) constraint

$$R^2 = L_1^2 + L_2^2 + 2L_1 L_2 \cos\theta_2. \tag{6}$$

Substituting (6) into the manipulability expression (3) yields a form that depends only on the link lengths:

$$w(L_1, L_2) = L_1 L_2 \sqrt{1 - \left(\frac{R^2 - L_1^2 - L_2^2}{2L_1 L_2}\right)^2}. \tag{7}$$

This expression is included for completeness, as it corresponds to the full two-dimensional grid search over (L_1, L_2) before the problem is reduced using the circular reachability constraint.

Reduced one-dimensional sweep. Since the reachability condition given by Eq. (4) restricts feasible designs to lie on a circular locus in the (L_1, L_2) plane (for $L_1, L_2 > 0$), the morphology can be parameterized by a single variable:

$$L_1 = R\cos\phi, \qquad L_2 = R\sin\phi, \qquad \phi \in [0, \tfrac{\pi}{2}]. \tag{8}$$

Substituting (8) into (3) gives the normalized manipulability

$$\bar{w}_{\text{norm}} = |\sin(2\phi)|. \tag{9}$$

A sweep over ϕ confirms that the maximum occurs at

$$\phi = 45^\circ \quad \Rightarrow \quad L_1 = L_2 = R/\sqrt{2},$$

which matches the link-length solution obtained analytically. The corresponding joint angle $\theta_2 = 90^\circ$ follows from the analytical maximization in Section 3.1 and is shown here only for completeness, since it is not recovered from the one-dimensional sweep itself. The resulting configuration is visualized in Fig. 2.

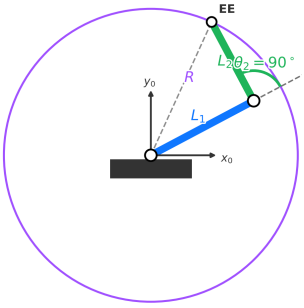

Fig. 2. Maximum-manipulability configuration of the planar 2R manipulator for a centered circular end-effector path ($L_1 = L_2 = R/\sqrt{2}$, $\theta_2 = 90^\circ$).

3.3 Heuristic Baselines

Before introducing the reinforcement learning approach, three commonly used black-box optimization methods are considered for comparison: Particle Swarm Optimization (PSO) (Kennedy and Eberhart (1995)), Bayesian Optimization (BO) (Mockus (1978); Jones et al. (1998)), and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Ostermeier (2006)). Each method searches over the link lengths (L_1, L_2) in order to maximize the average normalized manipulability along the specified end-effector path.

PSO is a population-based method in which a swarm of particles explores the design space and updates candidate solutions by combining personal best and global best information. BO constructs a probabilistic surrogate of the objective function and selects new samples through an acquisition rule that trades off exploration and exploitation. CMA-ES maintains a multivariate Gaussian distribution over the design variables and adapts its mean and covariance iteratively, enabling efficient sampling in regions of high performance.

These methods work well for the present 2-D circular morphology problem but require many objective evaluations and repeated inverse-kinematics calls. Their computational cost grows rapidly when additional morphology parameters are introduced (e.g., link offsets, joint limits, payload effects) or when the task is uncertain or time-varying (Zhang et al. (2008); De Luca and Oriolo (1992); Gros and Ferreau (2020)).

For this reason, RL is considered as a scalable alternative: once trained, a policy can output morphology parameters directly, without re-running an optimizer for every new task instance. In our experiments, PSO, BO, and CMA-ES are evaluated only on the circular case, where the analytical optimum is known and provides a ground-truth reference. The hyperparameters used for these heuristic baselines are listed in Table B.1.

4. REINFORCEMENT LEARNING FRAMEWORK

4.1 Problem Formulation

The morphology–optimization task is formulated as a single-step reinforcement learning (RL) problem. Since no temporal evolution is involved, the setting reduces to a *contextual bandit*, in which a geometry is selected and an immediate reward is returned. The state s represents the task path (e.g., circular, elliptical, or rectangular), and the action corresponds to the manipulator geometry,

$$a \;=\; [L_1,\, L_2,\, \theta_2]^\top \;\in\; \mathcal{A} \subset \mathbb{R}^3. \tag{10}$$

It is noted that this full 3-D action space is required only for non-circular tasks, where no closed-form optimum exists and manipulability depends jointly on (L_1, L_2, θ_2). For the circular task, the action dimensionality is reduced to a single parameter ϕ, since

$$L_1 = R\cos\phi, \qquad L_2 = R\sin\phi, \qquad \theta_2 = 90^\circ$$

are already known from the analytical solution. In that case, the RL agent selects only ϕ, and the task becomes a one-dimensional bandit.

The objective of learning is expressed as

$$J(\pi) \;=\; \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi(\cdot\,|\,s)}\big[r(s,a)\big], \tag{11}$$

where $\pi(\cdot|s)$ denotes the policy that outputs the geometry for a given task context (Sutton and Barto (2018)).

Notation for (11).

- $s \in \mathcal{S}$: task context (path descriptor). In this work it encodes the path type and parameters (e.g., circle radius R, ellipse (a_e, b_e), or rectangle (w, h)).
- $a \in \mathcal{A}$: action (geometry). For non-circular tasks, $a = [L_1, L_2, \theta_2]^\top$. For the circular task, the action reduces to a single parameter $a = [\phi]$ with $L_1 = R\cos\phi$, $L_2 = R\sin\phi$, and $\theta_2 = 90^\circ$.
- $\mathcal{D}$: distribution over task contexts used during training (in our experiments, one context per experiment; see Section 4.2 for how the path is sampled to compute rewards).
- $\pi(\cdot|s)$: (stochastic) policy over actions given context s, parameterized by neural networks (SAC/DDPG/PPO) and implemented with tanh-squashed outputs mapped to physical ranges. For DDPG the executed action is deterministic with added exploration noise during training.
- $r(s,a)$: scalar reward defined in Section 4.2 (one of r_{raw}, r_{norm}, r_{band}, or r_{hyb}).

The expectation in (11) is taken over $s \sim \mathcal{D}$ and $a \sim \pi(\cdot|s)$ (policy stochasticity and, when applicable, exploration noise) (Sutton and Barto (2018)).

4.2 Reward Design

The reward is computed by evaluating reachability and manipulability over N uniformly sampled points on the task path, $P = \{(x_k, y_k)\}_{k=1}^{N}$. Let $r_k = \sqrt{x_k^2 + y_k^2}$ denote the radial distance of the k-th point from the base, and define the morphology radii

$$r_{\min} = |L_1 - L_2|, \qquad r_{\max} = L_1 + L_2.$$

A point is considered reachable if it lies within this annulus and the elbow-up inverse kinematics returns a valid joint configuration within limits, as illustrated in Fig. 3:

$$\mathbb{I}_k = \begin{cases} 1, & r_k \in [r_{\min}, r_{\max}] \text{ and elbow-up IK} \\ & \text{returns } (\theta_{1,k}, \theta_{2,k}) \text{ within limits} \\ 0, & \text{otherwise.} \end{cases}$$

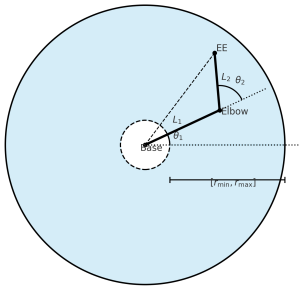

Fig. 3. Reachable annulus of the planar 2R manipulator showing link lengths L_1, L_2, joint angles θ_1, θ_2 (elbow-up), and the inner/outer radii $[r_{\min}, r_{\max}]$. The shaded band corresponds to the geometric annulus that is reachable in principle for the chosen (L_1, L_2).

The number of reachable samples and the corresponding coverage are

$$N_{\text{reach}} = \sum_{k=1}^{N} \mathbb{I}_k, \qquad \text{cov} = \frac{N_{\text{reach}}}{N}.$$

For all reachable targets ($\mathbb{I}_k$=1), the absolute and normalized manipulability values are

$$w_k = \left|L_1 L_2 \sin\theta_{2,k}\right|, \qquad w_k^{\text{n}} = \left|\sin\theta_{2,k}\right|,$$

and their averages are

$$\bar{w} = \frac{1}{N_{\text{reach}}} \sum_{k=1}^{N} \mathbb{I}_k \, w_k, \qquad \bar{w}_{\text{n}} = \frac{1}{N_{\text{reach}}} \sum_{k=1}^{N} \mathbb{I}_k \, w_k^{\text{n}},$$

with both set to zero if $N_{\text{reach}} = 0$.

Meaning of terms used above.

- $\boldsymbol{N}$: number of task samples; P is the set of sampled Cartesian targets.
- $\boldsymbol{r_k}$: radial distance of sample k; used to check geometric reachability.
- $\boldsymbol{r_{\text{min}}, r_{\text{max}}}$: inner/outer radii of the manipulator annulus induced by (L_1, L_2).
- $\boldsymbol{\mathbb{I}_k}$: binary reachability indicator for sample k (1 if reachable, 0 otherwise).
- $\boldsymbol{N_{\text{reach}}}$: number of reachable targets, i.e. $\sum_k \mathbb{I}_k$.
- **cov**: coverage ratio N_{reach}/N, measuring what fraction of the task is reachable.
- $\boldsymbol{\theta_{2,k}}$: elbow angle from IK for sample k (elbow-up convention).
- $\boldsymbol{w_k}$: absolute manipulability at sample k, incorporating link-scale and configuration.
- $\boldsymbol{w_k^{\text{n}}}$: normalized manipulability (configuration only, scale removed).
- $\boldsymbol{\bar{w}, \bar{w}_{\text{n}}}$: mean values of w_k and w_k^{n} over all reachable targets.

Band constraints. Each task is associated with a desired radial band $[b, a]$:

- circle (radius R): $[b, a] = [R, R]$,
- ellipse (semi-axes a_e, b_e): $[b, a] = [\min(a_e, b_e), \max(a_e, b_e)]$,
- rectangle (width w, height h): $[b, a] = \left[\frac{1}{2}\min(w, h), \frac{1}{2}\sqrt{w^2 + h^2}\right]$.

Band violation is penalized by

$$B \;=\; W_{\text{in}}\,[\,b - r_{\text{min}}\,]_+^2 \;+\; W_{\text{out}}\,[\,r_{\text{max}} - a\,]_+^2,$$

where $[x]_+ = \max(x, 0)$ ensures that penalties apply only when the annulus is too small or too large. Additional penalties are defined as

$$\Delta_{\text{unr}} = W_{\text{unr}}(1 - \text{cov}), \qquad \Delta_{\text{len}} = W_{\text{len}}(L_1 + L_2).$$

(Default weights are given in Table 1.)

Reward functions.

$$\begin{aligned} r_{\text{raw}} &= \bar{w} \;\; - \Delta_{\text{unr}} - \Delta_{\text{len}}, \\ r_{\text{norm}} &= \bar{w}_{\text{n}} - \Delta_{\text{unr}} - \Delta_{\text{len}}, \\ r_{\text{band}} &= -B \;\; - \Delta_{\text{unr}} - \Delta_{\text{len}}, \\ r_{\text{hyb}} &= \bar{w}_{\text{n}} - B - \Delta_{\text{unr}} - \Delta_{\text{len}}. \end{aligned}$$

Interpretation of penalties and rewards.

- $\boldsymbol{\Delta_{\text{unr}}}$ **(Coverage penalty)**: proportional to the fraction of unreachable points. This term discourages morphologies that leave large portions of the task unreachable; its weight W_{unr} sets how strongly infeasible designs are penalized.
- $\boldsymbol{\Delta_{\text{len}}}$ **(Length regularizer)**: penalizes large total link length $(L_1 + L_2)$. This avoids trivial solutions that artificially increase manipulability by enlarging both links; W_{len} controls the regularization strength.
- $\boldsymbol{B}$ **(Band penalty)**: enforces that the morphology annulus $[r_{\text{min}}, r_{\text{max}}]$ matches the task radial band $[b, a]$. The inner term $W_{\text{in}}[b - r_{\text{min}}]_+^2$ penalizes annuli that are too small, while the outer term $W_{\text{out}}[r_{\text{max}} - a]_+^2$ penalizes those that are too large.
- $\boldsymbol{r_{\text{raw}}}$: rewards absolute manipulability $\bar{w}$ while penalizing unreachability and overly long links. It is useful when absolute dexterity (scale included) matters, e.g., the centered circular task.
- $\boldsymbol{r_{\text{norm}}}$: rewards normalized manipulability $\bar{w}_{\text{n}}$ to emphasize configuration-dependent dexterity while removing scale effects; useful when link scale should not be preferred.
- $\boldsymbol{r_{\text{band}}}$: ignores manipulability and prioritizes geometric feasibility (coverage and band matching); useful when strict adherence to the task band is required.
- $\boldsymbol{r_{\text{hyb}}}$: combines normalized manipulability with coverage and band penalties, providing a balanced objective for non-circular tasks. This hybrid reward is used in the experiments where manipulability must be traded off against reachability and geometry.

The weight hyperparameters and algorithm-specific settings are given in Table 1.

4.3 Learning Algorithms

Soft Actor–Critic (SAC) (Haarnoja et al. (2018)), Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al. (2016)), and Proximal Policy Optimization (PPO) (Schulman et al. (2017)) are employed as representative state-of-the-art continuous-control algorithms. SAC and DDPG are trained off-policy using replay buffers and target critics, whereas PPO is trained on-policy using clipped surrogate objectives. All policies are implemented as multilayer perceptrons (MLPs) with tanh outputs, which are linearly mapped to the physical geometry ranges $L_i \in (L_{\text{min}}, L_{\text{max}})$ and $\theta_2 \in (0, \pi)$. [1]

For the circular task, the analytical optimum is known and the action can be reduced to the single parameter ϕ (i.e., $L_1 = R\cos\phi$, $L_2 = R\sin\phi$, $\theta_2 = 90°$). All three agents reliably recover the closed-form optimum, confirming the correctness of the RL setup.

For the elliptical and rectangular tasks, no closed-form solution exists and the full 3-D action space (L_1, L_2, θ_2) is used. In these cases, the hybrid reward r_{hyb} is adopted to balance manipulability with band feasibility, and the policies learn morphologies that maximize dexterity while satisfying geometric constraints. The algorithm-specific hyperparameters and reward weights are summarized in Table 1.

[1] Actions in $(-1, 1)$ are squashed by tanh and then scaled to their physical limits in the implementation.

Table 1. Hyperparameters used for reinforcement-learning algorithms (SAC, DDPG, PPO). Reward weights are shared across algorithms.

Parameter	**SAC**	**DDPG**	**PPO**
Episodes	5000	5000	5000
Batch size	256	256	128
Learning rate (actor/critic)	3×10^{-4}	1×10^{-3}	3×10^{-4}
Discount factor γ	0.99	0.99	0.99
Replay buffer size	10^5	10^5	—
Policy update ratio (PPO)	—	—	0.2
Target smoothing τ	0.005	0.005	—
Entropy coefficient α (SAC)	0.2	—	—
Optimizer	Adam	Adam	Adam
Reward weights (shared):			
W_{unr} (unreachability)		5.0	
W_{in} (inner-band)		5.0	
W_{out} (outer-band)		5.0	
W_{len} (length regularizer)		0.5	

5. RESULTS

5.1 Experimental setup: circle (analytical case)

For a centered circle of radius R, we use a scalar action $u \in (-1, 1)$ mapped to $\phi \in (\varepsilon, \frac{\pi}{2} - \varepsilon)$ via $\phi(u) = \frac{\pi}{4} + (\frac{\pi}{4} - \varepsilon)u$, with $\varepsilon = 10^{-3}$ rad. The link lengths are $L_1 = R\cos\phi$, $L_2 = R\sin\phi$, yielding the analytical normalized reward $w_{\text{norm}}(\phi) = \sin(2\phi)$ (no band penalty). The analytic maximizer is $\phi^\star = 45°$ with $L_1 = L_2 = R/\sqrt{2}$ and $w_{\text{norm}}^{\max} = 1$.

Table 2. Settings for the circle benchmark (one degree of freedom).

Item	**Value**
Path geometry	Circle of radius $R = 0.40\,\text{m}$ (centered at base)
Action/param	Scalar $u \in (-1, 1)$, mapped to $\phi(u)$ with $\varepsilon = 10^{-3}$
Lengths	$L_1 = R\cos\phi$, $L_2 = R\sin\phi$
Reward	$w_{\text{norm}} = \sin(2\phi)$ (no band penalty)
Analytic ref.	$\phi^\star = 45°$, $L_1 = L_2 = \frac{R}{\sqrt{2}}$, $w_{\text{norm}}^{\max} = 1$
Training	5 seeds; 5000 episodes; batch 256

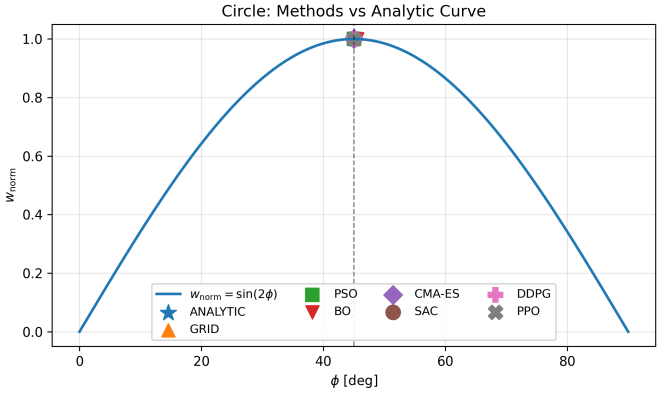

Fig. 4. Circle (analytical reward): method endpoints vs. analytical curve $w_{\text{norm}} = \sin(2\phi)$. Dashed: $\phi = 45°$.

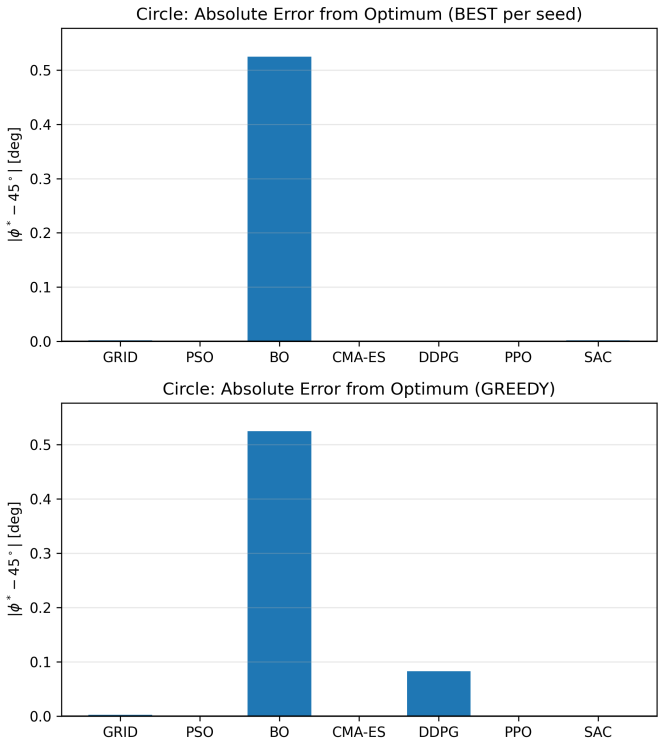

Fig. 5. Circle (analytical reward): absolute deviation $|\phi^\star - 45°|$ (deg). Top: BEST; Bottom: GREEDY.

Table 3. Circle benchmark: summary across methods

Method	$\phi^\star$ [deg]	L_1 [m]	L_2 [m]	w_{norm}
ANALYTIC	45	0.28284	0.28284	1
GRID	44.998	0.28285	0.28283	1
PSO	45	0.28284	0.28284	1
BO	45.525	0.28024	0.28542	0.999832
CMA-ES	45	0.28284	0.28284	1
DDPG	45	0.28284	0.28284	1
PPO	44.999	0.28285	0.28284	1
SAC	44.999	0.28285	0.28284	1

5.2 Experimental setup: ellipse/rectangle

Table 4. Settings used for ellipse/rectangle.

Item	**Value**
Path geometry	Ellipse: $a_e = \mathbf{0.40}\,\text{m}$, $b_e = \mathbf{0.25}\,\text{m}$ / Rectangle: $w = \mathbf{0.70}\,\text{m}$, $h = \mathbf{0.40}\,\text{m}$
Sampling/IK	$N = \mathbf{720}$ points, elbow-up
Action bounds	$L_i \in [\mathbf{0.05}, \mathbf{0.60}]\,\text{m}$ (via tanh box)
Band targets	Ellipse: $[b, a] = [\min(a_e, b_e), \max(a_e, b_e)]$; Rectangle: $[b, a] = [\frac{1}{2}\min(w, h), \frac{1}{2}\sqrt{w^2 + h^2}]$
Baselines	*Equal-dex:* $L_1 = L_2 = L^\star$, $L^\star = \sqrt{\mathbb{E}[r^2]/2}$, with $\mathbb{E}[r^2]_{\text{ellipse}} = \frac{1}{2}(a_e^2 + b_e^2)$ and $\mathbb{E}[r^2]_{\text{rect}} = \frac{w^3 + 3wh^2 + 3hw^2 + h^3}{12(w+h)}$ (derivations in App. A). *Band-match:* for band $[b, a]$, take $L_1 = \frac{a+b}{2}$, $L_2 = \frac{a-b}{2}$ (either order).
Seeds/Training	Seeds $= \{1, 2, 3, 4, 5\}$; episodes $= \mathbf{5000}$; batch $= \mathbf{256}$

Across ellipse and rectangle, we use a shared protocol and the hybrid reward (Sec. 4.2). Band targets follow geometry; actions are box–constrained to $L_i \in [0.05, 0.60]$ m via tanh; $N = 720$ path points are sampled with elbow–up IK.

Ellipse: results

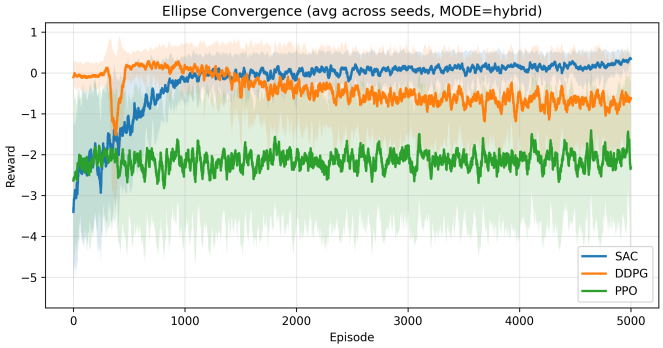

Fig. 6: Ellipse: training curves averaged across seeds.

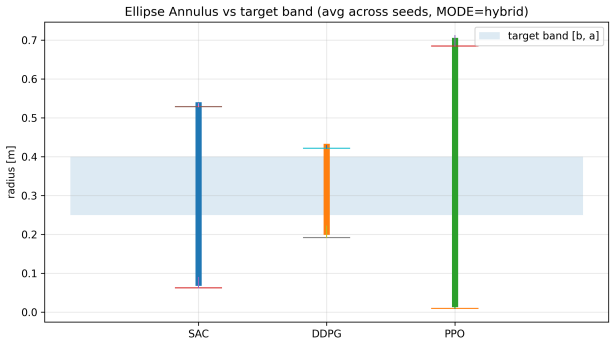

Fig. 7: Ellipse (hybrid reward): learned reachable annulus $[r_{\min}, r_{\max}]$ averaged across seeds for SAC, DDPG, and PPO, compared against the target task band $[b, a]$ (shaded). Vertical bars show the learned inner/outer radii; small horizontal ticks indicate the across-seed variation.

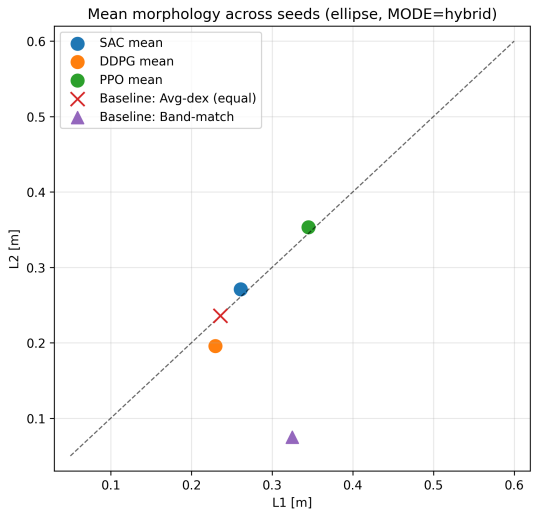

Fig. 8: Ellipse (hybrid reward): mean morphology across seeds. Dashed: $L_1 = L_2$.

Figure 7 complements the mean-length plot in Fig. 8 by showing how closely each algorithm matches the target band $[b, a]$. DDPG produces the tightest annulus, with $r_{\min}$ and $r_{\max}$ aligned near the band edges. SAC tends to overshoot the outer radius and slightly undershoot the inner radius, yielding a wider annulus than required but with stable behavior across seeds. PPO exhibits the largest spread, with a very small $r_{\min}$ and a large $r_{\max}$, indicating a conservative morphology that spans well beyond the desired band under the same 5,000-episode budget.

These annulus patterns are consistent with the training curves in Fig. 6: DDPG reaches higher rewards early but shows large fluctuations, which explains the larger variance across seeds in the final geometries. SAC learns more slowly but converges smoothly, leading to more consistent morphologies even if the solution is slightly less band-optimal. PPO shows the slowest and lowest reward convergence, which correlates with its broader and less task-focused annulus in Fig. 7. Together, the learning curves and the final annuli confirm that DDPG is more exploitative but less stable, whereas SAC offers a steadier trade-off between reward and geometric feasibility.

Table 5: Ellipse (hybrid): per-algorithm summary (mean ± std across seeds).

Metric	SAC	DDPG	PPO
L_1 [m]	0.261±0.043	0.229±0.115	0.345±0.013
L_2 [m]	0.271±0.043	0.196±0.114	0.353±0.016
Reward	0.43±0.031	0.629±0.011	−0.237±0.053

Rectangle: results

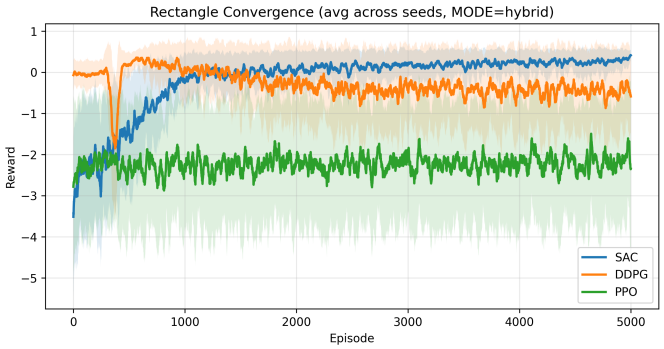

Fig. 9: Rectangle: training curves averaged across seeds.

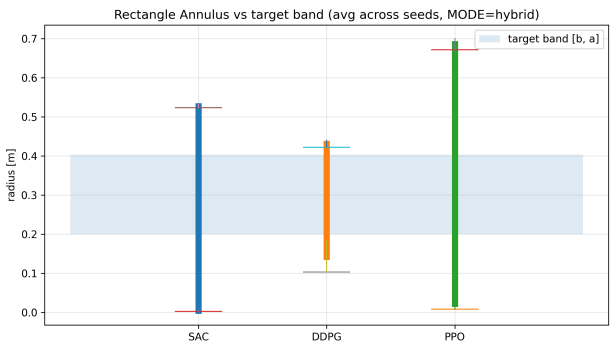

Fig. 10: Rectangle (hybrid reward): learned reachable annulus $[r_{\min}, r_{\max}]$ averaged across seeds for SAC, DDPG, and PPO, compared against the target task band $[b, a]$ (shaded). Vertical bars show learned inner/outer radii; horizontal ticks denote across-seed variation.

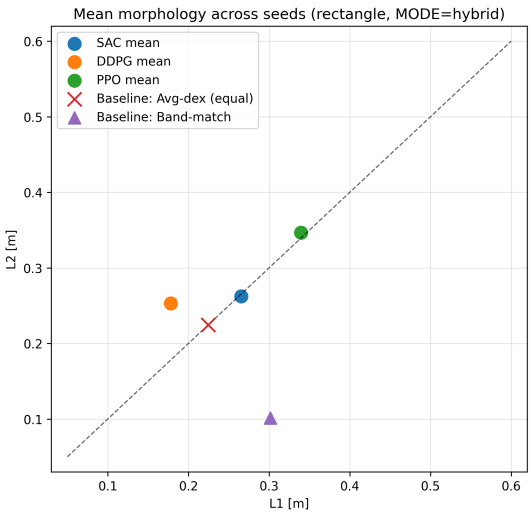

Fig. 11: Rectangle (hybrid reward): mean morphology across seeds. Dashed: L_1=L_2.

Figure 10 shows the same analysis for the rectangular task. As in the ellipse case, DDPG produces the narrowest annulus and stays closest to the desired band, although with noticeable variation across seeds. SAC again overshoots the outer radius and undershoots the inner radius, resulting in a wider annulus that still overlaps the target region. PPO yields the largest spread and tends to cover a much larger radial range than required, reflecting its lower sample efficiency under the same 5,000-episode budget.

These annulus outcomes align with the training curves in Fig. 9: DDPG reaches high reward early but with strong oscillations, which explains why its final geometries vary more from seed to seed. SAC improves more steadily and converges smoothly, leading to more consistent morphologies even if the final annulus is not the tightest. PPO again shows the slowest reward growth and lowest final return, which matches its tendency to produce conservative, overextended morphologies in Fig. 10. Overall, the rectangular results reinforce the same pattern observed in the ellipse task: DDPG is fast but unstable, SAC is slower but reliable, and PPO is the least sample-efficient in this setting.

Table 6: Rectangle (hybrid): per-algorithm summary (mean ± std across seeds).

Metric	SAC	DDPG	PPO
L_1 [m]	0.265±0.004	0.178±0.073	0.34±0.014
L_2 [m]	0.262±0.003	0.253±0.07	0.347±0.018
Reward	0.379±0.011	0.612±0.014	−0.114±0.059

6. DISCUSSION AND FUTURE SCOPE

The results above show that a simple contextual-bandit RL loop can recover analytical optima and provide competitive morphologies for non-analytical tasks. This section outlines practical scalability considerations, possible extensions and directions for future work that would increase the scope, realism and applicability of the proposed framework.

- **Higher–DoF morphology.** The formulation can be extended from a planar 2R manipulator to an n–DoF manipulators design by collecting the design variables into an action vector $a = [L_1, \ldots, L_n, \psi]$ and using manipulability $w(\theta) = \sqrt{\det(JJ^\top)}$ with $J \in \mathbb{R}^{m\times n}$. Exhaustive search grows exponentially with n, while the RL/bandit approach scales with the action dimension; the dominant per-episode cost remains the IK/Jacobian evaluation over N samples (roughly $O(N\,n^2)$). This cost is amenable to parallelization across samples and seeds, but careful algorithmic choices and dimensionality reduction (e.g., shared encoders or structured parameterizations) are required for large n.
- **Richer objectives and constraints.** The current work focuses on kinematic dexterity, but actuator limits and dynamics can be incorporated by adding penalties or constraints to the reward. For example, joint-velocity limits follow from $\dot\theta = J^{-1}\dot x$ and torque limits from $\tau = M(\theta)\ddot\theta + C(\theta,\dot\theta)\dot\theta + g(\theta)$; penalties can aggregate maximum or mean violations along the path. Alternative isotropy measures (e.g., condition number $\kappa(J)$ or task-aligned manipulability) and accuracy/stiffness objectives can be included without changing the learning loop, yielding more physically realistic designs.
- **From paths to regions and 3D tasks.** The radial "band" idea generalizes naturally by sampling targets from 2-D or 3-D workspace distributions rather than a single path. Orientation tracking can be handled by the $6 \times n$ spatial Jacobian, with reward terms that weight translation and rotation according to task priorities. This extension enables design for volumetric workspaces and full SE(3) tasks.
- **Sample efficiency and generalization.** Improving data efficiency is crucial for higher complexity. Promising approaches include curricula (loose → tight bands), task randomization over shape/size, shared encoders for task context, offline pretraining on synthetic task distributions, and differentiable-kinematics fine-tuning. Such measures improve transfer across tasks and reduce the number of expensive online evaluations.
- **Deployment and robustness.** Real-world deployment requires modeling uncertainty in L_i, actuator backlash, sensing noise, and fabrication tolerances. Robust objectives — for example optimizing expected performance or tail risk (CVaR) — and validation via a URDF pipeline or hardware-in-the-loop tests are natural next steps to bridge sim-to-real gaps.
- **Multi-objective design.** Practical design problems often trade off dexterity, coverage, energy consumption and mass. Instead of producing a single optimum, the framework can be extended to return Pareto fronts (via scalarization, multi-objective RL, or Pareto RL) so practitioners obtain families of morphology candidates that expose trade-offs for downstream selection.

Taken together, these directions sketch a roadmap from the present kinematic study towards a comprehensive, dynamic, and robust morphology co-design pipeline suitable for more complex robots and real-world validation.

7. CONCLUSION

We studied morphology optimization for planar manipulators using an analytical, heuristic, and reinforcement-learning (RL) toolkit. For the centered circle, the closed-form optimum ($L_1{=}L_2{=}R/\sqrt{2}$, $\theta_2{=}90°$) is recovered numerically by RL and heuristics, matching the analytical reward $w_{\text{norm}}(\phi){=}\sin(2\phi)$. For ellipse and rectangle, we introduced a hybrid reward function that balances normalized dexterity with geometric feasibility via a band penalty. Across all the algorithms (SAC, DDPG, and PPO), the agent attains competitive or superior performance to classic search baselines under identical evaluation budgets, with SAC typically offering the most robust greedy solutions and DDPG learning fastest but with higher variance.

The main takeaway is that a single, simple contextual-bandit RL loop can handle non-analytical tasks and constraint trade-offs that are cumbersome for grid search or black-box heuristics, while remaining consistent with analytical ground truth when it exists. The scope of the work is limited to planar kinematics considering absence of explicit torque/velocity, collision, and orientation objectives. Future work can scale to higher-DoF and 3-D tasks, incorporate dynamic constraints and multi-objective criteria, and validate on hardware with robust/sim-to-real training.

REFERENCES

Craig, J.J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson, Upper Saddle River, NJ, 3rd edition.

Farooq, S.S. et al. (2021). Optimal design of tricept parallel manipulator with particle swarm optimization. *ISA Transactions*, 113, 111–121.

Gros, S. and Zanon, M. (2020). Numerical optimal control for nonlinear systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 85–115.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor–critic: Off–policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, 1861–1870.

Hansen, N. (2006). The CMA evolution strategy: A comparing review. In J.A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea (eds.), *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, 75–102. Springer, Berlin.

Jones, D.R., Schonlau, M., and Welch, W.J. (1998). Efficient global optimization of expensive black–box functions. *Journal of Global Optimization*, 13(4), 455–492.

Kennedy, J. and Eberhart, R.C. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, 1942–1948.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. *International Conference on Learning Representations*.

Luca, A.D. and Siciliano, B. (1992). Closed-form dynamic equations of planar manipulators with flexible links. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5), 885–894.

Medvet, G., Bartoli, A., and Gustin, S. (2021). Optimizing the sensory apparatus of voxel-based soft robots through evolution and babbling. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 134–142. ACM. doi:10.1145/3449639.3459377.

Mockus, J., Tiesis, V., and Zilinskas, A. (1978). The application of bayesian methods for seeking the extremum. In L.C.W. Dixon and G.P. Szegő (eds.), *Towards Global Optimization*, volume 2, 117–129. Elsevier.

Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Shahbazi, M. et al. (2024). Optimization of dynamic parameter design of stewart manipulators using particle swarm optimization. *AIP Conference Proceedings*, 2548, 030024.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. Springer, London.

Stroppa, F., D'Ambrosio, D.B., Berretti, S., Buongiorno, M., Scioni, E., and Prattichizzo, D. (2024). Optimizing soft robot design and tracking with and without evolutionary computation: An intensive survey. *Robotica*, 42(8), 2848–2884. doi:10.1017/S0263574724001079.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition.

Weisstein, E.W. (n.d.). Ellipse and rectangle. `https://mathworld.wolfram.com/`. From *MathWorld–A Wolfram Web Resource*.

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2), 3–9.

Zhang, D. and Song, Y. (2008). Performance comparison of optimization techniques for robot design. *Mechanism and Machine Theory*, 43(1), 1–13.

Appendix A. DERIVATIONS OF $\mathbb{E}[r^2]$

The following expressions for the mean squared radius $\mathbb{E}[r^2]$ of the ellipse and rectangle are obtained from standard analytic-geometry integrations of $r^2 = x^2 + y^2$ along their respective perimeters (Weisstein (n.d.)).

Ellipse. With $x = a\cos t$ and $y = b\sin t$, the radial distance is $r^2 = a^2\cos^2 t + b^2\sin^2 t$. Uniformly averaging over $t \in [0, 2\pi]$ gives

$$\mathbb{E}[r^2]_{\text{ellipse}} = \frac{1}{2\pi}\int_0^{2\pi} (a^2\cos^2 t + b^2\sin^2 t)\, dt = \tfrac{1}{2}(a^2 + b^2).$$

Rectangle. For an axis-aligned rectangle of width w and height h, centered at the origin, the perimeter-weighted average $\mathbb{E}[r^2] = \frac{1}{2(w+h)}\int_{\partial\mathcal{R}}(x^2 + y^2)\, ds$ evaluates to

$$\mathbb{E}[r^2]_{\text{rect}} = \frac{w^3 + 3wh^2 + 3hw^2 + h^3}{12(w + h)}.$$

These results are consistent with the geometric formulations summarized in (Weisstein (n.d.)).

Appendix B. HEURISTIC OPTIMIZER SETTINGS

Table B.1. Heuristic optimizer settings for 1-D ϕ (maximize $f(\phi) = \sin(2\phi)$).

	PSO	**BO (GP+EI)**	**CMA-ES**
Evals (total)	3630	45	720
Iters	120	40	60
Pop / batch	n_p=30	n_{init}=5	λ=12, μ=6
Seed	0	0	0
Init sampling	$U[\phi_{\min}, \phi_{\max}]$	$U[\phi_{\min}, \phi_{\max}]$	$m_0 = \frac{\phi_{\min}+\phi_{\max}}{2}$
Init scale	$v \sim U[-0.05, 0.05]$	—	$\sigma_0 = 0.20(\phi_{\max} - \phi_{\min})$
Update core	$v \leftarrow \omega v + c_1 r_1 (p-x) + c_2 r_2 (g-x)$	GP (RBF ℓ=0.12, noise 10^{-10}) + EI (ξ=0.01), grid 2000	rank-μ; σ clip $[10^{-4}, 0.5(\phi_{\max}-\phi_{\min})]$
Params	ω=0.72 c_1=c_2=1.49	lengthscale 0.12	elitist
Bounds	clip to $[\phi_{\min}, \phi_{\max}]$	grid inside bounds	clip to bounds