

Kinematic Invariant Parametric Snake Robot Gait Design for Enceladus Exploration

Jack Naish

University of Cambridge
jrhn2@cam.ac.uk

Jacob Rodriguez, Jenny Zhang

Massachusetts Institute of Technology
{jacobrod, jenzhang}@mit.edu

Andrew Orehov, Devansh Dhrafan, Howie Choset

Robotics Institute, Carnegie Mellon University
{aorekhov, ddhrafan, choset}@andrew.cmu.edu

Guglielmo Daddi

Politecnico di Torino, Italy
guglielmo.daddi@polito.it

Bryson Jones, Rob Royce, Michael Paton, Masahiro Ono, Rohan Thakker

Jet Propulsion Laboratory, California Institute of Technology
{bryson.jones, robert.r.ristine, michael.paton, rohan.a.thakker}@jpl.nasa.gov

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

Snake-inspired robots demonstrate versatility through challenging terrains. However, their high-dimensional continuous action spaces make analytical gait design difficult. Early pioneers showed gait parameterization over low-dimensional spatially and temporally varying sine waves can serve as basis functions in gait shape space. Yet, optimizing these gaits is a non-convex problem as parameter space is fraught with local optima exacerbated by constraints such as actuator limits. Reinforcement Learning (RL) has emerged as a promising alternative for gait search. However, end-to-end RL approaches lack the safety guarantees necessary for deep space missions and don't yield an interpretable representation of the learned gait. We propose a hybrid method that first identifies sidewinding gaits for novel bend-twist kinematic chains using RL, before distilling the gait into an equivalent open-loop parametric approximation. Our method avoids the twist-windup and twist-jump problems (key shortcomings we identified with prior work) while combining task-space learning with an interpretable policy. Simulation and hardware experiments demonstrate our proposed method can generate parametric gaits for a simple line-following task where an existing optimization-based curve-fitting method cannot.

Keywords: reinforcement learning, snake robotics, gait design, parametric curve fitting.

1. Introduction

From exploring icy crevasses on distant moons [Tempest et al. \(2020\)](#) to traversing the trachea [Boehler et al. \(2020\)](#), snake robots represent a promising mode of mechanized mobility. These systems typically consist of actuated links chained together in series. Compared to traditional locomotion systems such as rovers or drones, the hyper redundancy of snake robots coupled with their minimal cross-sectional area makes them well suited to operating in extreme environments. Yet, characterized by their high degree of freedom and complex terra-mechanical interactions, snake robots are challenging to control. The motion of one module necessarily affects all others, leading to a tightly coupled non-linear system.

The promise of snake robots has prompted intense research, continually pushing these systems towards higher degrees of autonomy and versatility [Sato et al. \(2010\)](#); [Kano and Ishiguro \(2013\)](#); [Ito and Fukumori \(2006\)](#); [Ye et al. \(2007\)](#); [Duivon et al. \(2022\)](#). One such system at the NASA Jet Propulsion Laboratory is the Exo-biology Extant Life Surveyor (EELS, fig. 1(a)). Measuring 4m and weighing 100kg, this early technology demonstrator aims to one day seek out signs of life on Enceladus, a distant moon of Saturn [Vaquero et al. \(2024\)](#). EELS consists of ten modules, each fitted with a propulsive screw, and chained together via novel two-axis bend-twist joints.

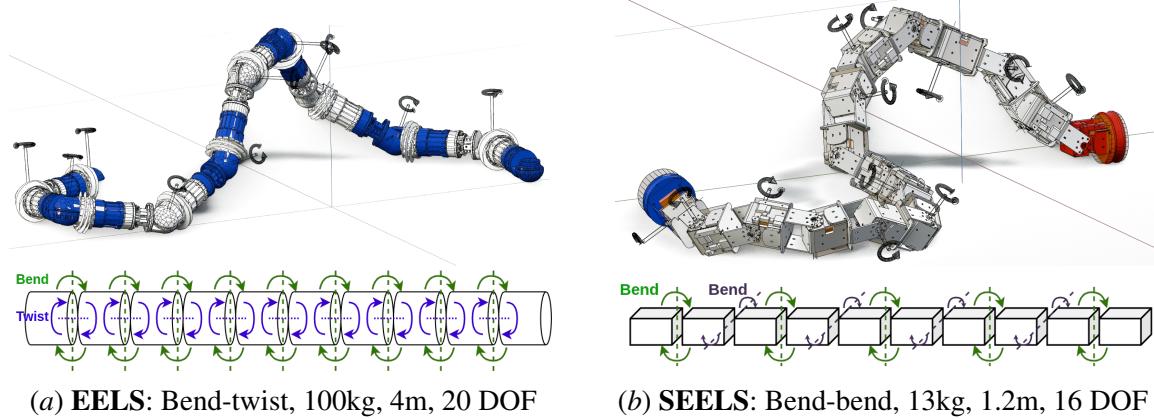


Figure 1: The two kinematic configurations considered in this study.

The bend-twist joints of EELS (fig. 1(a)) are advantageous compared to status-quo orthogonally arranged bend-bends, like those on SimpleEELS (SEELS, fig. 1(b)). Bend-twists reduce the effective length of each module by combining two joints into one, yield the necessary torque density for sub-surface locomotion, and increase internal module space for scientific payloads [Gildner et al. \(2024\)](#). However, EELS twist joints have maximum deflection hard stops at $\pm 2\pi$. This introduces a phenomenon we term *twist-windup*, where roll in successive twists tends to accumulate, reach the limit, and stall the gait. Thus, designing bend-twist gaits is a challenging research problem, both for EELS and the community at large.

2. Background

The space of possible gaits for even a simple snake robot is immense, broadly categorized into *sidewinding*, *concertina*, *rectilinear*, and *pipe-crawling* classes. Early approaches to serpentine locomotion were first pioneered by [Hirose \(1994\)](#). Through propagating parameterized traveling waves down a continuous backbone curve, the undulating motion of a side-winding desert snake was emulated, allowing for coupled translational and rotational control.

$$\phi_{\text{bend}}(n, t) = \begin{cases} \beta_e + A_e \sin(\omega_s n + \omega_t t) & \text{if } n \text{ is even} \\ \beta_o + A_o \sin(\omega_s n + \omega_t t + \delta) & \text{if } n \text{ is odd} \end{cases} \quad (1)$$

Variations on this idea have been thoroughly explored. The serpentoid curve [Shi et al. \(2016\)](#); [Tang et al. \(2018\)](#); [Gong et al. \(2013\)](#) is a common heuristic function that operates over combinations of spatial ω_s and temporal ω_t phase shifts, joint offsets δ , and reference angles β to describe motion. Manual gait design strategies like this necessitate empirically tuned thresholds derived through expert observation. Nevertheless, once parameters are found, the method performs well on bend-bend

configurations [Zade et al. \(2021\)](#). Unfortunately, application to EELS is complicated by bend-twist joints, invalidating the effectiveness of such a simple approach.

Optimization approaches to gait design represent the current class of state-of-art techniques. A common theme is so-called *curve-fitting* whereby a gait is defined in high-level task space, then converted into a corresponding low-level joint space. [Xiao et al. \(2015\)](#) approached this via *Locomotion Reduction*, which projects the problem into an embedding space by taking a previously formulated basis gait, such as *conical sidewinding*, and substituting a steering control law. This reduces the system to an approximation of the simpler differential-drive car model.

Generalization by hierarchically decomposing gait design has also been explored. [Hatton and Choset \(2010\)](#) introduced *Annealed Chain Fitting* (ACF) to translate a continuous backbone curve into joint angles, and *Keyframe Wave Extraction* to identify parameterized periodic functions that produce those sequences. Previously, this method had been successfully applied to torsion-free kinematic chains where the robot can bend but not twist with respect to itself. When we attempted to generalize ACF to our bend-twist case (in Section 4), either the optimizer failed to fit the kinematic chain to the curve while satisfying twist constraints (twist windup), or it converged to an alternate kinematic assembly mode that would require rapid changes in actuator angles (twist jumps).

These two failure modes motivate our use of reinforcement learning for a more robust, complete gait search through joint hyperspaces. Relevant techniques include proximal Policy Optimization [Yongqiang et al. \(2021\)](#) [Bing et al. \(2020a\)](#), Deep Q-learning [Zhang et al. \(2024\)](#), and Policy Fitting [Wang et al. \(2021\)](#), which to date have all been applied in an end-to-end manner. These works focus on improvements in energy efficiency and movement speed compared to classical methods like the Serpentoid equation introduced above.

However, RL in snake robotics tends to exploit several small, but consequential, simplifications. The frictional anisotropy of snakeskin is often used to justify imposing side-slip constraints, typically achieved by installing passive wheels, skates, or tracks along each module. Furthermore, to the best of our knowledge, prior work in this area exclusively focuses on bend-bend configurations, likely because orthogonally arranged joints simplify mechanical design. Both of these assumptions are invalidated on EELS, which precludes artificially imposed sideslip constraints due to its disabled propulsive screws, and has a bend-twist configuration.

More importantly, end-to-end RL methods lack the safety guarantees required for robust deployment in risk-critical settings, such as during deep space missions, despite recent emphasis on sim-to-real transfer [Bing et al. \(2022\)](#). This is because actions are stochastically sampled from a probability distribution. The interpretability and introspection offered by a simple gait equation is also lost with RL since the behavior of learned control policies is implicitly represented by the weights of a neural network.

3. Methodology

A two-fold approach (fig. 2) is proposed to address the previously established shortcomings. First, to mitigate non-convexity challenges introduced by twist constraints in bend-twist kinematic configurations, we cast gait optimization as a Markov Decision Process solved using RL. This is done without artificially imposed frictional anisotropy. Once trained, the policy is frozen, an episode roll-out executed, and joint trajectories recorded. To address interpretability and safety guarantees, a parametric representation is fitted to the joint trajectories by either 1) approximation via the top k dominant gait frequency components or 2) applying least squares regression between the gait's joint

trajectory and classical serpentoid equation to automatically infer its coefficients. We then evaluate the two parametric forms across simulations and a hardware experiment, showing our method works with bend-twist and bend-bend chains.

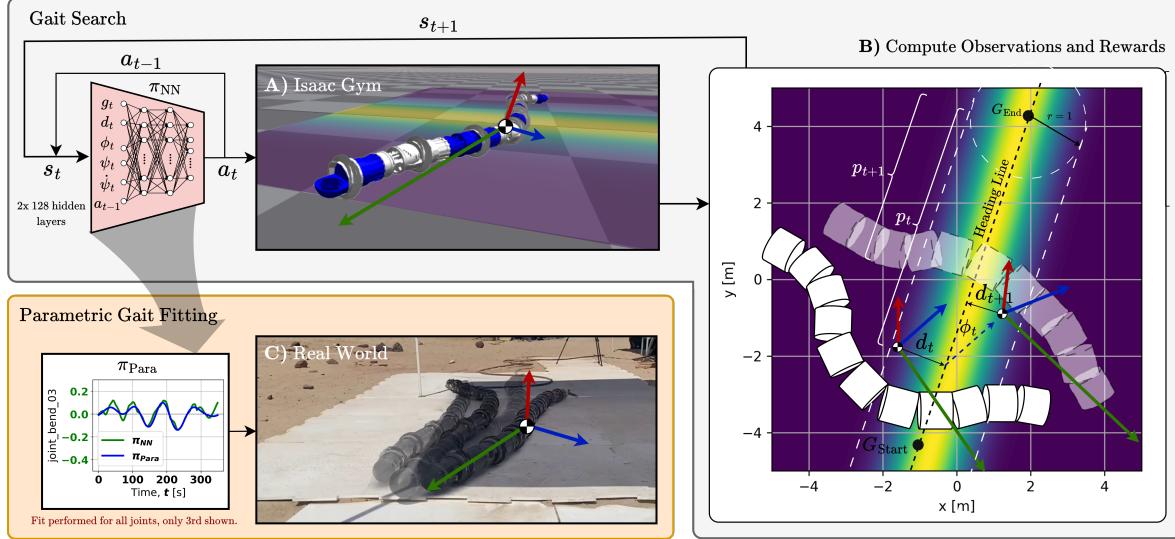


Figure 2: **Methodology overview.** (A) shape-based locomotion policy is trained in simulation to move from a start point to goal using RL. (B) Progress rewards incentivize reaching the goal while safety rewards (gradient) penalize lateral divergence. An action regularisation term avoids twist windup. (C) The policy is reconstructed into an equivalent parametric form and replayed on the real robot in an open-loop manner.

3.1. Gait Search

We begin by building a neural network controller capable of learning a shape-based gait that minimizes the time taken to reach a designated goal without prior knowledge of snake dynamics. To achieve this, EELS is simulated in Nvidia’s IsaacGym [Makoviychuk et al. \(2021\)](#) as a 20-degree-of-freedom bend-twist kinematic chain with uniform module mass and moments of inertia.

The shape-based locomotion problem is framed as an infinite-horizon Markov Decision Process defined by tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The agent starts from a state $s_t \in \mathcal{S}$, drawn from an initial state distribution. For each time step t , an action $a_t \in \mathcal{A}$ is sampled from a stochastic policy $\pi_\theta(a_t|s_t)$, with weights θ . a_t is executed, the agent transitions to new state $s_{t+1} \in \mathcal{S}$ and receives reward $r_{t+1} \in \mathcal{R}$ discounted by γ . The objective is to find network parameters θ that maximize the expected discounted reward. Off the shelf Proximal Policy Optimization from [RL_Games](#) [Makoviichuk and Makoviychuk \(2021\)](#) is used to find θ .

A reward function that minimizes time-to-target directly encapsulates our learning objective. However, this signal is sparse and can only be calculated upon the successful completion of an episode. Prior works have addressed this hurdle by designing dense *proxy rewards* to approximate an underlying task reward [Hadfield-Menell et al. \(2017\)](#). In car and drone racing contexts, projecting an agent’s body frame onto a curve and maximizing the distance between current and previous projected positions over successive timesteps has shown promising results [Song et al. \(2021\)](#); [Fuchs et al. \(2020\)](#).

Importantly, though, progress rewards rely on a body frame that is fixed to the chassis and which remains unaffected from internal shape changes. This requirement is trivially met on quadrotors. In the robotic snake domain, defining such a frame that intuitively describes the robot’s motion is harder. We avoid fixing this frame to a physical snake module (i.e., head, tail) like in prior work [Bing et al. \(2022\)](#) [Bing et al. \(2020b\)](#) as this will likely bias the types of learned gaits. Alternatively, constraining joint limits would impede the versatility of the shape-based performance envelope. [Rollinson and Choset \(2011\)](#) demonstrated that a body frame whose origin is the snake’s center of mass and whose axes are aligned with the snake’s principal moments of inertia is one where the intuitive notions of position and orientation prevail. This averaged frame, termed the *Virtual Chassis*, isolates the internal motion of the robot’s shape changes from its external motion. Figure 2 (B) depicts this idea by drawing center-of-mass indicators over two sequential iterations.

For theoretical details on the Virtual Chassis, we point the interested reader to [Rollinson and Choset \(2011\)](#). Here, the necessary results are restated. First, a $n \times 3$ data matrix is constructed with rows representing individual module locations in world space, shifted by the snake’s overall center of mass ($\bar{x}, \bar{y}, \bar{z}$) (eq. (3)). Next, the Single Value Decomposition (SVD) is applied, decomposing \mathbf{P} into three new matrices eq. (4). U and V are discarded. Importantly, S describes a rotation matrix that aligns the axes of the body frame with the principal moments of inertia of the module positions about the center of mass. To ensure uniqueness after the decomposition, the third singular vector of S is set as the cross product of the first and second. This Virtual Chassis frame accurately describes the average position and orientation of the snake.

$$\mathbf{p}_i = \begin{bmatrix} x_i - \bar{x} \\ y_i - \bar{y} \\ z_i - \bar{z} \end{bmatrix} \in \mathbb{R}^3 \quad (2) \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1^\top \\ \vdots \\ \mathbf{p}_n^\top \end{bmatrix} \quad (3) \quad \mathbf{P} = USV^\top \quad (4)$$

To formalize the line-following locomotion task, we draw a heading line between the start location of an episode G_{Start} and the end G_{End} . At time t , the distance between the virtual chassis projected onto the heading line, and the goal, is p_t . The shortest distance between the snake’s center of mass and the heading line is denoted d_t , as depicted in fig. 2 (B).

Rewards: Progress reward in our context are given by r_t^{prog} in eq. (6). Maximizing progress reward equates to maximizing the velocity of the virtual chassis towards the goal location, incentivizing getting there in minimal time. Early experiments trained using progress rewards alone revealed a tendency for the agent to diverge laterally from the heading line. Thus, *safety rewards* were introduced to minimize divergence d_t , keeping the virtual chassis centered. The gradient background of fig. 2 shows the safety reward over a 5x5m interval, and eq. (6) defines it symbolically. To prevent twist windup discussed earlier, an action penalty term r_t^{action} is applied to incentivize minimal deflection of the snake’s joints, keeping them near their zero point throughout a gait cycle.

$$r_t = r_t^{\text{prog}} + r_t^{\text{safe}} + r_t^{\text{action}} + r_t^{\text{heading}} \quad (5)$$

$$= \underbrace{\left(\|G_{\text{Start}} - p_{t-1}\| - \|G_{\text{End}} - p_t\| \right)}_{\text{Progress Reward}} + \underbrace{C_1 \exp\left(\frac{-d_t^2}{C_2}\right)}_{\text{Safety Reward}} - \underbrace{C_3 |a_t|^2}_{\text{Action Penalty}} + \underbrace{C_4 \exp\left(\frac{-\psi_t^2}{C_5}\right)}_{\text{Heading Reward}} \quad (6)$$

States: The state vector encodes observations made by the robot into a $3+3n$ length real vector, for n the number of snake modules eq. (7).

$$s_t = [p_t \ d_t \ \phi_t \ \psi_t \ \dot{\psi}_t \ a_{t-1}]^T \quad (7)$$

A heading angle, ψ_t , is formed between the heading line and the y (forward, blue) axis of the virtual chassis. ϕ_t is a vector of n snake joint angles (both bend and twist) with $\dot{\phi}_t$ their corresponding angular velocities. a_{t-1} is the policy action from the previous timestep. We justify privileged information, such as ground truth progress, heading, and divergence measurements, in our state vector since this dependency is removed via parametric policy reconstruction (Section 3.2).

Actions: Each action $a_t \in \mathbb{R}^n$ is a vector of desired module joint bend angles. Joints are articulated using proportional derivative (PD) controllers with desired joint position set-points commanded by the RL policy at each time step. The robot is terminated if it reaches the goal, the divergence distance d_t is greater than a fixed threshold, or the episode length is exceeded.

3.2. Parametric Gait Fitting

Once a stable line-following policy π_θ is learned using RL in simulation, policy weights θ are frozen, a single rollout is performed, and the resultant joint trajectories are recorded. This produces joint time-series like those shown in fig. 3(a) and fig. 3(b). To parametrically approximate the recording, Fast Fourier Transform (FFT) and Least Squares Regression (LSR) are applied. Each method assumes independence between individual joints.

$$\bar{\phi}(t, n) \approx \sum_{k=1}^K A_{n,k} \sin(2\pi\omega_{n,k}t + \delta_{n,k}) \quad (8) \quad \bar{\phi}(t, n) \approx A_n \sin(\omega_s n + \omega_t t + \delta_n) + \beta_n \quad (9)$$

Fast Fourier Transform: FFT is a common method for approximating functions as a sum of sinusoids. In the Snake domain, we acknowledge its prior use in dominant frequency identification in gait design [Gong et al. \(2014\)](#), particularly when inferring parametric gaits from recordings of biological snakes [Gong et al. \(2016\)](#). We take a similar approach, instead fitting to an RL policy rollout instead of a biological counterpart. The top K dominant frequency components of a joint trajectory are extracted using FFT with the trajectory is approximated as a sum of these components eq. (8). Larger K increases approximation fidelity at the expense of model complexity. K is ablated as a hyper-parameter in Section 4.

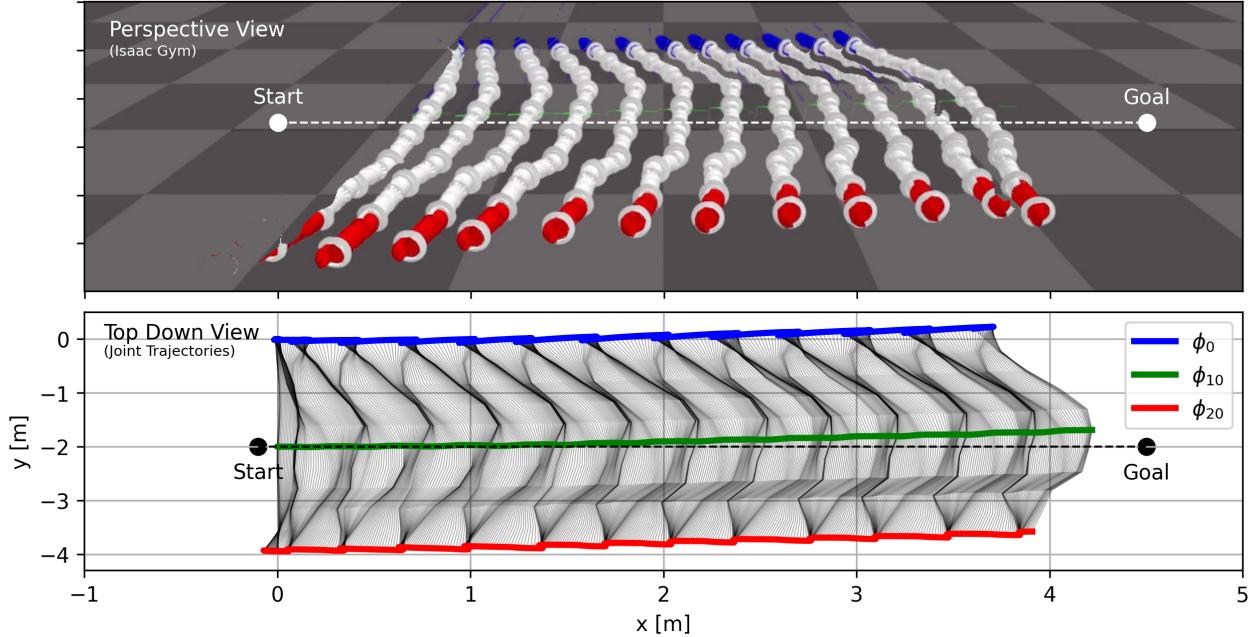
Least Squares Regression: For each joint, LSR fits the corresponding joint trajectory to the Serpentoid Equation eq. (9) by finding the equation coefficients ($A, \omega_t, \omega_s, \delta, \beta$) that minimize the sum of squared residuals. The `scipy.curve_fit` function is used [Gommers et al. \(2022\)](#).

4. Results & Discussion

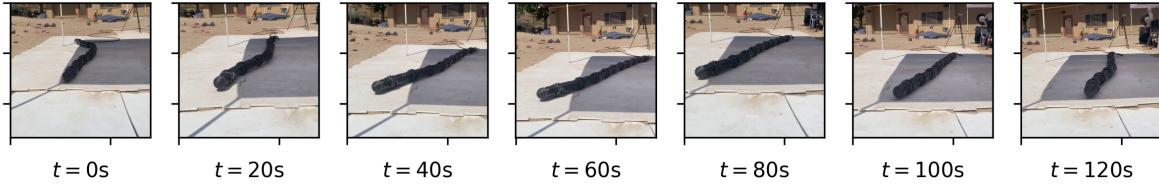
Our experiments aim to answer: 1) Can complexity introduced by twist joint constraints be avoided by casting gait optimization as an RL problem? 2) What is the translational distance and lateral divergence of RL, RL + FFT, and RL + LSR gaits during the line-following task? We ask both for bend-twist (EELS fig. 1(a)) and bend-bend (SimpleEELS fig. 1(b)) kinematics. Annealed Chain Fitting from [Hatton and Choset \(2010\)](#) is implemented as a baseline for EELS, and the classical Serpentoid equation for SimpleEELS.

Figure 3(a) demonstrates the first part of our method, learning a shape-based gait via RL to track a heading line. Minimal lateral divergence between the virtual chassis and line is observed. Trajectories for head (ϕ_0 , bend), middle (ϕ_{10} , twist), and tail (ϕ_{20} , bend) modules are highlighted. A complex time-varying periodic gait structure over 13 cycles is evident. This proved an emergent policy property since sinusoidal oscillations were not explicitly incentivized in the MDP.

Figure 3(b) shows the RL + FFT parametric gait deployed on EELS in the JPL Mars Yard. The robot traverses $\approx 0.4\text{m}$, illustrating our method’s potential for real world transfer.



(a) **Sim:** RL (closed-loop) Sidewinding policy that avoids twist windup while tracking heading line.



(b) **Real:** RL + FFT (open-loop) parametric reconstructed gait moving $\approx 0.4\text{m}$ in JPL Mars Yard.

Figure 3: Key result of our training and deployment process.

Figure 4 depicts FFT and LSR parametric approximations for the rollout given in fig. 3. Both parametric methods obtain acceptable reconstruction performance. FFT yields the closest approximation using a sum of $K = 75$ sines, translating 3.8m towards the goal in 600 timesteps. Conversely, LSR approximates the RL gait using a single sinusoidal component, at the expense of poor reconstruction quality early in the gait ($t < 200$). A marginally reduced translation distance (2.5m) results. Evidently, LSR can only fit the dominant gait frequency mode. While sufficient to capture the general shape, it can’t encode higher-frequency, more nuanced details.

The Annealed Chain Fitting (ACF) baseline (fig. 4, bottom) demonstrates a key failure mode where joint angles *windup* or rapidly *jump* during a rollout. Joint *jumps* are likely a consequence of the numerical solver rapidly switching between different local optima in joint hyperspace across keyframes while trying to optimise backbone fit. Such erratic changes in joint angles can’t be accurately tracked by the low-level PD controllers and if deployed in the real world, would damage the hardware. Clearly, while the path traced through joint hyperspace by ACF over successive keyframes is geometrically correct, it is not guaranteed to be practically feasible. As this *joint*

flipping failure mode was not reported by the original authors for torsion-free bend-bend configurations, we conclude it is a consequence of the increased kinematic complexity imposed by EELS’s twist joints.

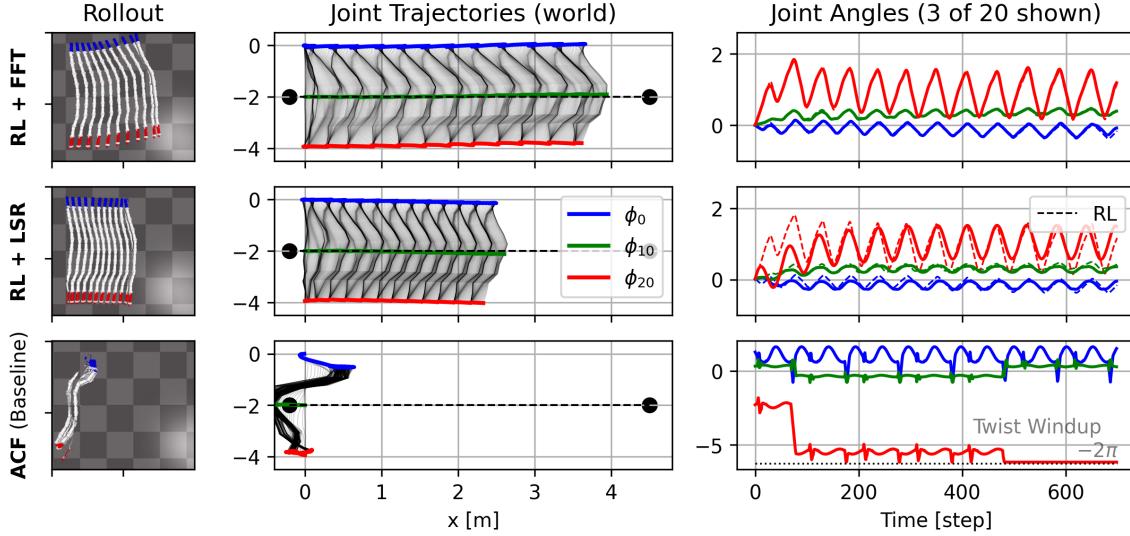


Figure 4: Bend-Twist Case. (**Top, Middle**): Our parametric methods (FFT, LSR) fitted to the RL rollout in fig. 3(a). (**Bottom**): Sidewinding Annealed Chain Fitting baseline from fig. 5.

Figure 5 further analyses why the ACF rollout shown in fig. 4 (bottom) fails. ACF begins by *sculpting* a bend-twist chain to a parametric backbone curve (left column, fig. 5). This is done via an iterative, sliding window numerical optimization process. Twist joints introduce a degree of freedom tangent to the curve, which surprisingly makes it easier for the solver to obtain a better fit compared to bend-bends. These fits are performed at discrete ‘keyframes’ over a fixed gait cycle (middle). Extracted Keyframe Waves are interpolated, producing the joint-angle plot (right column) representing commands sent to the robot. Actuator jumps and windup phenomena are annotated for a subset of three joints. Windup angles exceeding $\pm 2\pi$ are clamped, stalling that joint. Jumps cause rapid changes in joint angles, potentially damaging hardware.

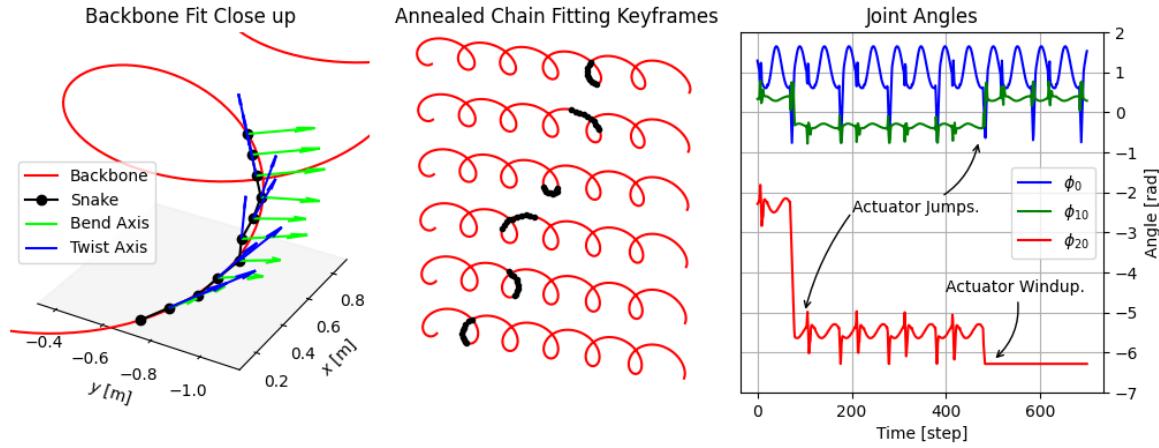


Figure 5: Baseline Experiment. (**Left**): Annealed Chain Fitting (Hatton and Choset (2010)) between bend-twist snake robot and side-winding backbone curve (**Middle**): Extracted Keyframes of annealed sidewinding gait. (**Right**): Gait joint angles during rollout.

Next, we show our method also works for the easier status-quo bend-bend case through application to SEELS (fig. 1(b)). We again train a sidewinding gait using RL (fig. 6 top) before fitting the FFT (upper middle) and LSR (lower middle) approximations to this single RL rollout. The Serpentoid baseline (bottom) obtains similar translation and divergence distances but at a much slower speed (its episode length was increased from $T = 600$ to $T = 3000$).

Our method reduces the burden of expert tuning associated with Serpentoid coefficients or backbone curves as gait search is offloaded to RL. Because the RL policy can explore a larger region of the state space during training (than manual empirical search), it tends to find policies that increase the effective use of snake actuators, yielding better and translation distances while minimising lateral divergence.

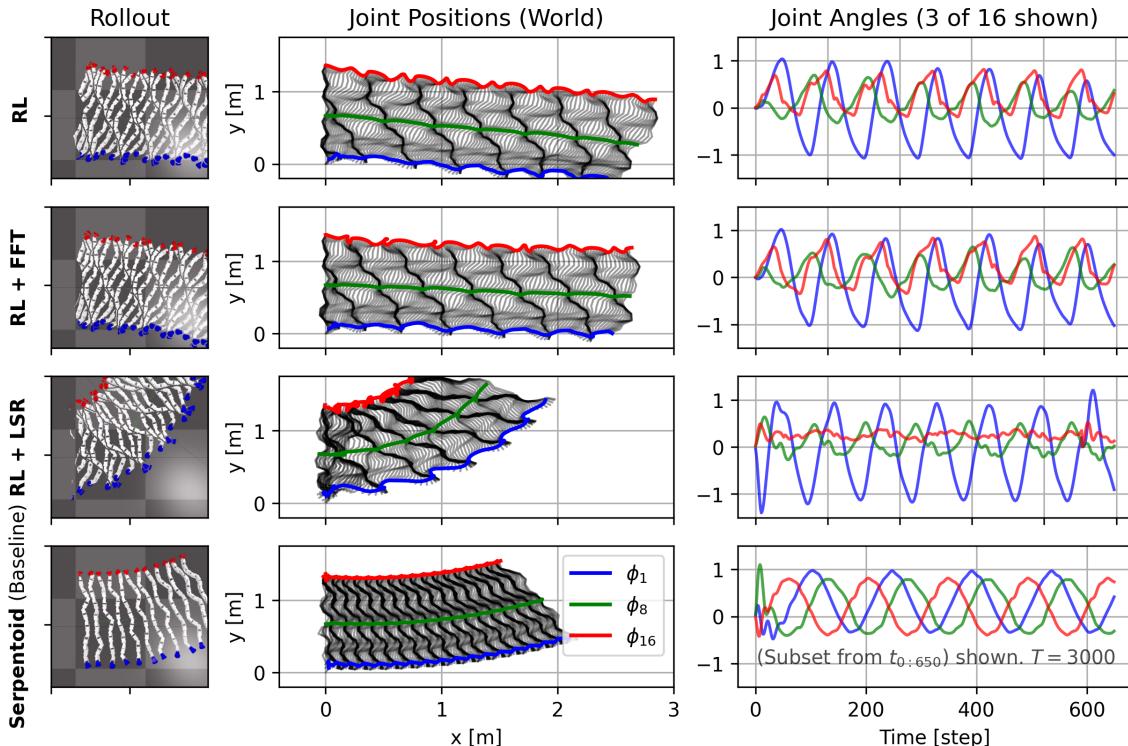


Figure 6: Bend-Bend Case. **(Top)**: RL Gait. **(Middle)**: FFT and LSR approximations to RL gait. **(Bottom)**: Manually tuned Serpentoid Equation baseline (eq. (1)).

Importantly, these results collectively show our method is also kinematic configuration invariant, capable of deriving gaits for both bend-twist (fig. 4) and bend-bend cases (fig. 6). To the best of our knowledge, this is a unique property. Specialised classical approaches such as ACF or the Serpentoid Equation explicitly assume a bend-bend design. However, this only guarantees kinematic transfer. A low-level tracking controller is still required for the dynamics to transfer for any general snake system.

Finally, we ablate the FFT frequency component count, k , which determines parametric approximation fidelity. We can principally select k by analysing the gait frequency distribution (fig. 7 left) and the effect of k on distance travelled compared to the parent RL gait (fig. 7 right). Evidently, Seels (blue line) has three distinct gait frequency peaks. Setting $k = 3$ proved sufficient to reconstruct its RL gait. Increasing $k > 3$ improves fit quality but with no notable improvement in

traversal distance beyond that learned by the RL gait. Conversely, for EELS, frequency-amplitude peaks are comparatively less pronounced, suggesting its learned gait is more complex as it uses a broader spectrum of component frequencies. This increases the requirement on k , with RL + FFT only achieving comparable traversal distances for $k \geq 75$. Evidently, there is a quality-of-fit versus complexity trade-off that must be considered when selecting k for RL+FFT.

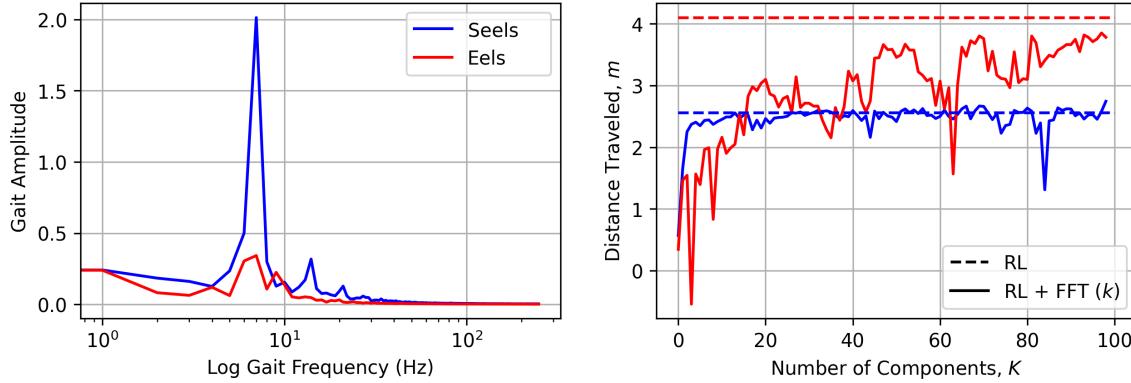


Figure 7: (**Left**): Gait frequency distribution. (**Right**): Ablation over k FFT components.

5. Conclusion & Future Work

The primary challenge when designing gaits for snake robots lies in their high-dimensional continuous joint spaces, compounded by the presence of bend-twist actuators. Until now, there has been no principled method to design parametric gaits for bend-twist snake robots. In this work, we proposed a hybrid approach to meet this need, surpassing the limitations of Annealed Chain Fitting and the Serpentoid Equation on novel bend-twist kinematic configurations.

Our method achieves larger translation distances with smaller lateral divergence on the line following task. This was achieved by casting gait design as an RL problem, avoiding challenging non-convexity by replacing hard twist constraints with a soft reward term. Through our parametric gait distillation process, the benefits of safety, determinism, and interpretability are preserved. The twist-jump and twist-windup failure modes of ACF are avoided. Kinematic invariance across two different chain configurations (bend-twist, bend-bend) is also demonstrated.

Gaits produced by our method can be used as building blocks for more complex behaviours by high-level mission planners further up the robotics control stack. We believe this work represents a small, but valuable, step towards enabling EELS to robustly and safely explore distant worlds.

Despite our promising results, several limitations are evident. The RL formulation is only capable of learning a single forward translation motion primitive. Future works could examine more sophisticated gaits for turning or translating over non-planar terrain. Our method also assumes kinematic chain symmetry. However, some snake robots have asymmetric modules, eg: a large sensor head at one end. Future work could explore head-holding constraints to avoid high-speed impacts damaging sensitive electronics. We also adopted a very simple FFT and LSR scheme for parametric gait approximation. Future studies could examine more sophisticated variations on this theme that yield better fits, possibly by incorporating a Gaussian Process with a specialized periodic kernel function.

6. Acknowledgements

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). Jack Naish acknowledges the generous support of the New Zealand Space Scholarship, the combined Cambridge Trusts-Rosalie Crawford Girton Scholarship, and the Pillman and Cody Fund.

References

- Zhenshan Bing, Christian Lemke, Long Cheng, Kai Huang, and Alois Knoll. Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning. *Neural Networks*, 129:323–333, September 2020a. ISSN 0893-6080. doi: 10.1016/j.neunet.2020.05.029.
- Zhenshan Bing, Christian Lemke, Fabric O Morin, Zhuangyi Jiang, Long Cheng, Kai Huang, and Alois Knoll. Perception-action coupling target tracking control for a snake robot via reinforcement learning. *Frontiers in Neurorobotics*, 14:591128, 2020b.
- Zhenshan Bing, Long Cheng, Kai Huang, and Alois Knoll. Simulation to real: Learning energy-efficient slithering gaits for a snake-like robot. *IEEE Robotics & Automation Magazine*, 29(4): 92–103, 2022.
- Quentin Boehler, David S. Gage, et al. REALITI: A Robotic Endoscope Automated via Laryngeal Imaging for Tracheal Intubation. *IEEE Transactions on Medical Robotics and Bionics*, 2(2): 157–164, January 2020. ISSN 2576-3202. doi: 10.1109/TMRB.2020.2969291.
- Askan Duivon, Pino Kirsch, et al. The Redesigned Serpens, a Low-Cost, Highly Compliant Snake Robot. *MDPI*, 2022. ISSN 2218-6581. URL <https://uia.brage.unit.no/uia-xmlui/handle/11250/3012088>.
- Florian Fuchs, Yunlong Song, et al. Super-Human Performance in Gran Turismo Sport Using Deep Reinforcement Learning. *arXiv*, August 2020. doi: 10.1109/LRA.2021.3064284.
- Matthew Gildner, Nikola Georgiev, Eric Ambrose, Torkom Pailevanian, Avak Archanian, Hovhannes Melikyan, Daniel Loret de Mola Lemus, Michael Paton, Rohan Thakker, and Masahiro Ono. To boldly go where no robots have gone before—part 2: The versatile mobility of the eels robot for robustly exploring unknown environments. In *AIAA SCITECH 2024 Forum*, page 1965, 2024.
- Ralf Gommers, Pauli Virtanen, Evgeni Burovski, Warren Weckesser, Travis E Oliphant, Matt Haberland, David Cournapeau, Tyler Reddy, Pearu Peterson, Andrew Nelson, et al. *scipy/scipy: Scipy 1.9. 0. Zenodo*, 2022. URL https://web.archive.org/web/20240223013146/https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html.
- Chaohui Gong, Matthew J. Travers, Xiaozhou Fu, and Howie Choset. Extended gait equation for sidewinding. In *2013 IEEE International Conference on Robotics and Automation*, pages 5162–5167. IEEE, May 2013. doi: 10.1109/ICRA.2013.6631315.

- Chaohui Gong, Matthew Tesch, et al. Snakes on an inclined plane: Learning an adaptive sidewinding motion for changing slopes. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 14–18. IEEE, 2014. doi: 10.1109/IROS.2014.6942697.
- Chaohui Gong, Matthew J Travers, Henry C Astley, Lu Li, Joseph R Mendelson, Daniel I Goldman, and Howie Choset. Kinematic gait synthesis for snake robots. *The International Journal of Robotics Research*, 35(1-3):100–113, 2016.
- Dylan Hadfield-Menell, Smitha Milli, et al. Inverse Reward Design. *Advances in Neural Information Processing Systems*, 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/32fdab6559cdfa4f167f8c31b9199643-Abstract.html.
- Ross L. Hatton and Howie Choset. Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction. *Auton. Robot.*, 28(3):271–281, April 2010. ISSN 1573-7527. doi: 10.1007/s10514-009-9175-2.
- Shigeo Hirose. Biologically Inspired Robots: Snake-Like Locomotors and Manipulators by Shigeo Hirose Oxford University Press, Oxford, 1993, 220 pages, incl. index (£40). *Robotica (Cambridge. Print)*, 1994. URL <https://www.semanticscholar.org/paper/Biologically-Inspired-Robots%3A-Snake-Like-Locomotors-Owen/b84dfcd1184c570118c9ab77c335bc15863e531f>.
- K. Ito and Y. Fukumori. Autonomous control of a snake-like robot utilizing passive mechanism. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 15–19. IEEE, 2006. ISBN 978-0-7803-9505. doi: 10.1109/ROBOT.2006.1641741.
- Takeshi Kano and Akio Ishiguro. Obstacles are beneficial to me! Scaffold-based locomotion of a snake-like robot using decentralized control. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 03–07. IEEE, 2013. doi: 10.1109/IROS.2013.6696821.
- Denys Makoviichuk and Viktor Makoviychuk. rl_games. *Github*, May 2021. URL https://github.com/Denys88/rl_games. [Online; accessed 10. May 2024].
- Viktor Makoviychuk, Lukasz Wawrzyniak, et al. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. *arXiv*, August 2021. doi: 10.48550/arXiv.2108.10470.
- David Rollinson and Howie Choset. Virtual chassis for snake robots. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 221–226. IEEE, September 2011. doi: 10.1109/IROS.2011.6094645.
- Takahide Sato, Wataru Watanabe, and Akio Ishiguro. An adaptive decentralized control of a serpentoid robot based on the discrepancy between body, brain and environment. In *2010 IEEE International Conference on Robotics and Automation, ICRA 2010*, pages 709–714. 2010. doi: 10.1109/ROBOT.2010.5509236.
- Peipei Shi, Qianjun Shao, and Dongtai Liang. Design and improved serpentoid curve locomotion control of a planar modular snake robot. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 1398–1402. IEEE, August 2016. doi: 10.1109/ICInfA.2016.7832038.

Yunlong Song, Mats Steinweg, et al. Autonomous Drone Racing with Deep Reinforcement Learning. *arXiv*, March 2021. doi: 10.48550/arXiv.2103.08624.

Lei Tang, Li-Min Zhu, Xiangyang Zhu, and Guoying Gu. A serpentoid Curve Based Motion Planning Method for Cable-Driven Snake Robots. In *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pages 1–6. IEEE, November 2018. doi: 10.1109/M2VIP.2018.8600874.

Marco Tempest, Hiro Ono, et al. EELS ICRA Presentation. *Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration*, 2020, May 2020. URL <https://ntrs.nasa.gov/citations/20220001316>.

T. S. Vaquero, G. Daddi, R. Thakker, M. Paton, A. Jasour, M. P. Strub, R. M. Swan, R. Royce, M. Gildner, P. Tosi, M. Veismann, P. Gavrilov, E. Marteau, J. Bowkett, D. Loret de Mola Lemus, Y. Nakka, B. Hockman, A. Orekhov, T. D. Hasseler, C. Leake, B. Nuernberger, P. Proen  a, W. Reid, W. Talbot, N. Georgiev, T. Pailevanian, A. Archanian, E. Ambrose, J. Jasper, R. Etheredge, C. Roman, D. Levine, K. Otsu, S. Yearicks, H. Melikyan, R. R. Rieber, K. Carpenter, J. Nash, A. Jain, L. Shiraishi, M. Robinson, M. Travers, H. Choset, J. Burdick, A. Gardner, M. Cable, M. Ingham, and M. Ono. EELS: Autonomous snake-like robot with task and motion planning capabilities for ice world exploration. *Sci. Rob.*, 9(88), March 2024. ISSN 2470-9476. doi: 10.1126/scirobotics.adh8332.

Renpeng Wang, W. Xi, Xian Guo, and Yongchun Fang. Path Following for Snake Robot Using Crawler Gait Based on Path Integral Reinforcement Learning. *International Conference on Advanced Robotics and Mechatronics*, 2021. URL <https://www.semanticscholar.org/paper/Path-Following-for-Snake-Robot-Using-Crawler-Gait-Wang-Xi/6267b505a0bcd5161f31a054aaaf1566e9460a17>.

Xuesu Xiao, Ellen Cappo, et al. Locomotive reduction for snake robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 26–30. IEEE, 2015. doi: 10.1109/ICRA.2015.7139718.

Changlong Ye, Shugen Ma, et al. Development of a 3D Snake-like Robot: Perambulator-II. In *2007 International Conference on Mechatronics and Automation*, pages 05–08. IEEE, 2007. doi: 10.1109/ICMA.2007.4303526.

Qi Yongqiang, Yang Hailan, Rong Dan, Ke Yi, Lu Dongchen, Li Chunyang, and Liu Xiaoting. Path-integral-based reinforcement learning algorithm for goal-directed locomotion of snake-shaped robot. *Discrete Dynamics in Nature and Society*, 2021:1–12, 2021.

Harshad Zade, Aadesh Varude, Karan Pandya, Ajinkya Kamat, Shital Chiddarwar, and Rohan Thakker. Requibis-reconfigurable quadrupedal-bipedal snake robots. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 2241–2246. IEEE, 2021.

Dong Zhang, Renjie Ju, and Zhengcai Cao. Reinforcement learning-based motion control for snake robots in complex environments. *Robotica*, 42(4):947–961, April 2024. ISSN 0263-5747. doi: 10.1017/S0263574723001613.