

Multi-gait Locomotion Planning and Tracking for Tendon-actuated Terrestrial Soft Robot (TerreSoRo)

Arun Niddish Mahendran¹, Caitlin Freeman¹, Alexander H. Chang²,
Michael McDougall³, Patricio A. Vela² and Vishesh Vikas¹

Abstract—The adaptability of soft robots makes them ideal candidates to maneuver through unstructured environments. However, locomotion challenges arise due to complexities in modeling the body mechanics, actuation, and robot-environment dynamics. These factors contribute to the gap between their potential and actual autonomous field deployment. A closed-loop path planning framework for soft robot locomotion is critical to close the real-world realization gap. This paper presents a generic path planning framework applied to TerreSoRo (Tetra-Limb Terrestrial Soft Robot) with pose feedback. It employs a gait-based, lattice trajectory planner to facilitate navigation in the presence of obstacles. The locomotion gaits are synthesized using a data-driven optimization approach that allows for learning from the environment. The trajectory planner employs a greedy breadth-first search strategy to obtain a collision-free trajectory. The synthesized trajectory is a sequence of rotate-then-translate gait pairs. The control architecture integrates high-level and low-level controllers with real-time localization (using an overhead webcam). TerreSoRo successfully navigates environments with obstacles where path re-planning is performed. To best of our knowledge, this is the first instance of real-time, closed-loop path planning of a non-pneumatic soft robot.

I. INTRODUCTION

Since the advent of the soft McKibben actuator, soft materials have been envisioned to be an integral part of next generation robots, including for terrestrial environments. This can be attributed to their ability to adapt and interact with the environment. Over the past few decades there has been active research in discovery of novel soft materials, actuators and sensors [1], [2]. Correspondingly, there has been advancement in modeling and control techniques for soft systems like manipulators [3]. Additionally, researchers have demonstrated terrestrial locomotion of soft robots using ad-hoc or intuitive gaits [4]. However, in comparison to their rigid counterparts, path planning and navigation of soft robot locomotors is understudied and rarely implemented. This can be primarily attributed to challenges related to modeling of soft materials and their actuation, plus robot-environment interaction. Furthermore, unlike rigid robots, soft terrestrial robots generate lower magnitudes of force

when interacting with the environment. This has multiple consequences, including higher variance in locomotive gait displacements (translation and rotation) and higher sensitivity to small environmental changes. The research focus is to mitigate the sources of discrepancy between the potential of soft terrestrial robots and their real-life realization by developing a real-time, closed-loop locomotion controller with localization feedback.

Finite-element modeling and different reduced-order models have been explored by researchers for creating locomotion models [5]–[9]. However, the accuracy and predictability of these approaches remains limited. Models for the friction and sliding of soft materials over a substrate have proven inadequate to capture the robot-environment interaction. Additionally, soft robots are sensitive to manufacturing inaccuracies and defects. Consequently, most locomotion control strategies for soft terrestrial robots rely on biomimetic, intuitive approaches, or trial-and-error [4], [10]. More recently, environment-centric, data-driven model-free approaches have been implemented to synthesize gaits [11], [12]. This paper utilizes the gaits synthesized by these approaches as briefly discussed later in the paper.

Tethered and untethered soft locomotors are actuated using various methods, e.g., pneumatic, shape memory alloys (SMAs), dielectric elastomers (DEA), and motor-tendon actuators [4]. While most use open-loop control strategies, closed-loop control has been performed by few researchers. Patterson et al use a reactive strategy to perform closed-loop control of an SMA-actuated soft swimming robot [13]; Liu et al [14] use a reactive planner for control of a pneumatically actuated soft robot with predetermined gaits; Hamill et al [15] perform gait-based path planning using temporal logic; Lu et al [16] apply bidirectional A* with a time varying bounding box. For pose recovery, soft robotics researchers typically rely on economically expensive motion capture systems, e.g., VICON and Optitrack, limiting their more widespread study. This research employs a lattice-based trajectory planner; robot gait models inform the design of controlled trajectories that move the robot from start to goal while avoiding obstacles. It also details an experimental setup that uses two inexpensive overhead webcams and a localization algorithm that compensates for marker occlusion.

Multi-modal motion planning for traditional rigid-body mobile robots typically entails a solution search, through the robot configuration space, for a state sequence (and accompanying control) to accomplish an objective while respecting both task and feasibility constraints. Though artic-

1: The authors are with the Department of Mechanical Engineering, University of Alabama, Tuscaloosa, AL, USA. Email: anmahendran@crimson.ua.edu, clfreeman7@crimson.ua.edu, vvikas@ua.edu

2: The authors are with the Institute for Robotics and Intelligent Machines (IRIM) and School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Email: alexander.h.chang@gatech.edu, pvela@ece.gatech.edu

3: The author is with the University of Strathclyde Glasgow, Glasgow, UK. Email: michael.mcdougall5@gmail.com

This work was supported by NSF grants #1562911, #1830432

ulated mobile robots (e.g. humanoids, multi-legged mechanisms) are often natively described by high-dimensional configuration spaces, planning exploits dimensional-reduction strategies that specialize the search space to modes and configurations relevant to a particular task [17]–[19]; these manifest in conjunctions of several graph-based representations that together capture the multi-modal search space, and where sampling approaches may be utilized to construct graphs or graph-to-graph (i.e. mode-to-mode) transitions. Similarly, graph-based representations have been used to synthesize motion plans for vehicles capable of multiple geographically-dependent modalities (e.g. swimming, driving, flying); graphs are synthesized using a sampling-based approach, with edges valued according to modal cost of transport. Dijkstra’s algorithm may then be used to identify the optimal multi-modal solution when traveling between locations [20]. For a hyper-redundant snake-like robot, a hybrid-optimization approach employs mixed integer programming with model predictive control (MPC) to guide step climbing; the former enforces a particular sequence of discrete modes to be traversed, while the latter designs reference trajectories within each mode [21]. This work focuses on the TerreSoRo soft mobile robot, capable of several locomotion gaits; trajectory planning and re-planning entails a lattice-based search through the space of possible gait sequences, for guiding the robot to desired goal locations within obstacle-strewn environments and in the face of locomotion uncertainty.

Contributions. This research, to best of our knowledge, is the first instance of real-time path planning and closed-loop control of a non-pneumatic soft robot. The experimental setup involves system integration of high-level (online path planning and offline gait synthesis) control, low-level control (TerreSoRo actuation), and real-time pose estimation that uses parallel architecture to process visual feedback. Trajectory planning is accomplished using a lattice-based, greedy breadth-first search through the robot’s gait control space; motion models of available robot gaits inform the design of controlled locomotion trajectories that move the robot from start to goal while avoiding obstacles. A rotate-then-translate motion control paradigm is adopted to both simplify the procedure and aid tractability of the planning problem. The planned trajectory is re-computed when the position error between the estimated and the actual path exceeds a prescribed threshold.

Organization. Sec. II formulates the navigation problem, details the robot, and provides an overview of data-driven gait discovery and selection. Next, Sec. III discusses the closed-loop control framework, architecture and path-planning methodology. Sec. IV contains the experimental setup, methodology, tracking algorithms, and results. Sec. V concludes the paper and discusses future work.

II. ROBOT DESCRIPTION AND PROBLEM FORMULATION

A. TerreSoRo: Tetra-Limb Terrestrial Soft Robot

TerreSoRo is a four-limb terrestrial soft robot actuated using motor-tendon actuators (MTAs). The robot is powered

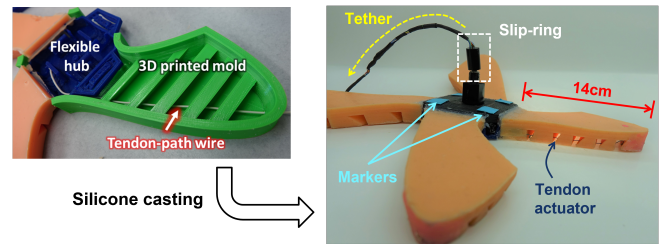


Fig. 1. The soft robot is cast using a rigid mold and a tendon-path wire. The motors with spools are placed inside the hub.

using external power with a low-level controller. The high-level controller and path planner is located off-board on a desktop computer that communicates with the webcam for localization (described in Sec. III). The robot design is the result of topology optimization to allow six identical robots to reconfigure into a sphere, whose details are more fully explored in [22], [23]. As a result, the limbs are designed for complex geometrical curling and not optimized for any particular locomotion modes. The emphasis is on implementing a path planning strategy for individual planar locomotion of the soft robot and does not address reconfigurability.

Robot fabrication involves integration of soft material limbs, control and actuation payload (motors, electronics), and routing of the tendons through the limb as shown in Fig. 1. The modular fabrication process involves mixing two liquid silicone components (Smooth-On Dragon Skin Part A and B - Shore Hardness 20A) degassed in vacuo. The tendon paths are cast by threading a thick wire through the rigid 3D printed mold as shown in Fig. 1 which is removed upon curing of the cast. The central hub is 3D printed with flexible filament (Shore Harness 85A, placed in the mold for casting; the casting is repeated for the other limbs. Rapid curling and uncurling of the flexible limbs (450 ms/transition) is achieved through motor-tendon actuation. Four DC motors with 3D printed PLA spools are placed in the hub and secured using zip ties. Teflon tubing is inserted into the individual fins of each limb to prevent tear caused by the difference in stiffness between the silicone and the fishing line tendon, Fig. 1. Finally, threaded fishing line attached to the spool is routed through each fin and anchored at the end with a fishing hook. A slip ring is incorporated into the tether connector to reduce any effects of built-up torsion in the tether.

B. Gait Synthesis

The gaits for TerreSoRo are synthesized (discovered) using an environment-centric framework that discretizes the factors dominating the robot-environment interaction. The procedure of synthesizing locomotion gaits is described briefly.¹

Environment-centric Framework. Conceptually, locomotion results from optimization of forces acting at different parts of the body that ultimately effect change in inertia [25].

¹The reader may refer to [11], [12] for detailed analysis, and to [24] for the graph theory terminology and concepts used.

In that context, *robot states* are defined as discrete physical states (e.g., postures, shapes) where the forces acting on the robot body in each state are considerably different. *Motion primitives* refer to the possible transitions between these robot states. These transitions result in motion of the robot (rotation and translation). A weighted digraph is effective in modeling the robot states, motion primitives, resulting motion, and their inter-dependencies. The robot states and motion primitives correspond to the digraph's n vertices $V(G)$ and the m directional edges $E(G)$, respectively. For TerreSoRo, robot states correspond to permutations of the four limbs being curled (actuated) or uncurled (un-actuated) as illustrated in Fig. 2. For this robot, actuation is binary (on/off) and all possible permutations (states) are statically stable. The weight associated with each edge e_i is the resulting motion of the motion primitive, i.e., the translation $\mathbf{p}_i \in \mathbb{R}^{2 \times 1}$ and rotation θ_i measured in the coordinate system of the initial vertex of the edge. For this discussion, the edge weight $\mathbf{w}_i$ is modeled as a normal distribution with mean $\boldsymbol{\mu}_i \in \mathbb{R}^{3 \times 1}$ and covariance matrix $\Sigma_i \in \mathbb{R}^{3 \times 3}$:

$$\mathbf{w}_i = \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i), \boldsymbol{\mu}_i(e_i) = \begin{bmatrix} \mathbf{p}_i \\ \theta_i \end{bmatrix}, \Sigma_i(e_i) = \begin{bmatrix} \Sigma_{pp} & \Sigma_{p\theta} \\ \Sigma_{\theta p} & \Sigma_{\theta\theta} \end{bmatrix}. \quad (1)$$

In summary, we define the following matrices with elements corresponding to edges e_i .

	Description	Dim
$P(E)$	mean displacement matrix, $P(e_i) = \mathbf{p}_i$	$\mathbb{R}^{2 \times m}$
$\Theta(E)$	mean rotation matrix, $\Theta(e_i) = \theta_i$	$\mathbb{R}^{1 \times m}$
$S_p(E)$	translation covariance trace matrix where $S_p(e_i) = \text{tr}(\Sigma_{pp}(e_i))$	$\mathbb{R}^{2 \times m}$
$S_\theta(E)$	rotation covariance matrix where $S_\theta(e_i) = \Sigma_{\theta\theta}(e_i)$	$\mathbb{R}^{1 \times m}$

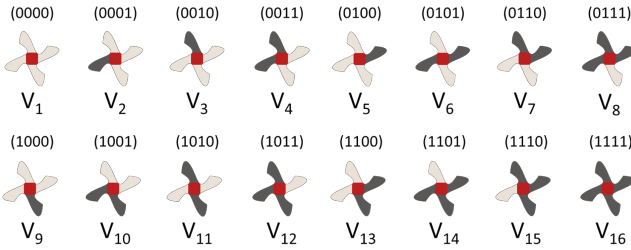


Fig. 2. The robot states represented by a four digit binary number. Each digit corresponds to one of the four limbs to indicate if the limb is actuated (1/ black color limb) or un-actuated (0/ light color limb).

Learning of the environment is equivalent to learning the graph edge weights. Experimentally, this is achieved by traversing all the edges of the graph without repeating any and recording the resulting motion. This traversal sequence, referred to as the Euler cycle, is repeated multiple (five) times with randomized starting positions and path orders to learn the probabilistic weights of the graph.

Translation and Rotation Gaits. Locomotion gaits are defined here as simple cycles that are transformation invariant. The transformation invariance principle implies that the

distance and rotation of the robot are preserved irrespective of starting vertex as it traverses through all edges of the simple cycle. It has been proven that under this definition, there will exist two types of planar gaits: translation and rotation [12]. The former is the gait when the cumulative rotation of the simple cycle is zero. The latter corresponds to the simple cycle when translation of all the edges is zero. Hence, we individually optimize for these two type of gaits with different cost functions and constraints.

The *gait library* comprises the synthesized translation and rotation gaits discovered using the discussed data-driven approach (learning of the graph and searching for optimal gaits). The cost functions, J_t, J_θ linearly weight the locomotion, variance and gait length while assuming small rotations of the motion primitives.

$$J_t(\mathbf{z}) = (\alpha_t^T P + \beta_t S_p + \gamma_t \mathbf{1}_{1 \times m}) \mathbf{z} \quad (2)$$

$$J_\theta(\mathbf{z}) = (\alpha_\theta \Theta + \beta_\theta S_\theta + \gamma_\theta \mathbf{1}_{1 \times m}) \mathbf{z}$$

where $\{\alpha_t, \beta_t, \gamma_t, \alpha_\theta, \beta_\theta, \gamma_\theta\}$ are the linear weights, and the binary vector $\mathbf{z} \in \{0, 1\}^m$ is the mathematical representation of a gait. Consequently, the gait synthesis is formulated as a Binary Integer Linear Programming (BILP) optimization problem with linear constraints that can be solved using optimization solvers, e.g., MATLAB®. The translation $\mathbf{z}_t$ and rotation $\mathbf{z}_\theta$ gaits are synthesized using

$$\mathbf{z}_t = \min_{\mathbf{z}} J_t(\mathbf{z}) \quad \text{s.t.} \quad \Theta \mathbf{z} \leq \epsilon_\Theta$$

$$\mathbf{z}_\theta = \min_{\mathbf{z}} J_\theta(\mathbf{z}) \quad \text{s.t.} \quad |\mathbf{Pz}| \leq \epsilon_p \forall z_i = 1 \quad (3)$$

Gait constraints: $B\mathbf{z} = 0, B^i \mathbf{z} \leq 1, z_i \in \{0, 1\} \forall i,$
 $\nexists \mathbf{z}_1, \mathbf{z}_2 \quad \text{s.t.} \quad \mathbf{z} = \mathbf{z}_1 + \mathbf{z}_2, B\mathbf{z}_1 = B\mathbf{z}_2 = 0$

where B is the incidence matrix, B^i is the positive elements of B , and the gait constraints mathematically ensure that vector $\mathbf{z}$ is a simple cycle.

Once the gait library has been created and contains a synthesized rotation gait and translation gait, symmetry of the robot can be assumed to expand the library to improve path planning capabilities. The permutations of the translation gait w.r.t. each limb are considered for control purposes to achieve change in orientation of 90 degrees. Such behavior is observed in biology (e.g., brittle star) [26] where the animal can change their leading limb to change the direction of their translation. While it is assumed that these permutations will have similar motion in four different directions, each of these gaits are tested to characterize their distinct twists.

C. Problem Statement

Soft mobile robots, such as TerreSoRo, manipulate their deformable body to accomplish distinctly useful locomotion, relative to more traditional rigid-body robots. In particular, locomotion results from unique interactions between their soft body material composition and the surrounding environment. This allowance comes with distinct challenges; locomotion outcomes are heavily coupled to manufacturing and material variabilities, factors that often are difficult to control. Data-driven techniques are demonstrably effective

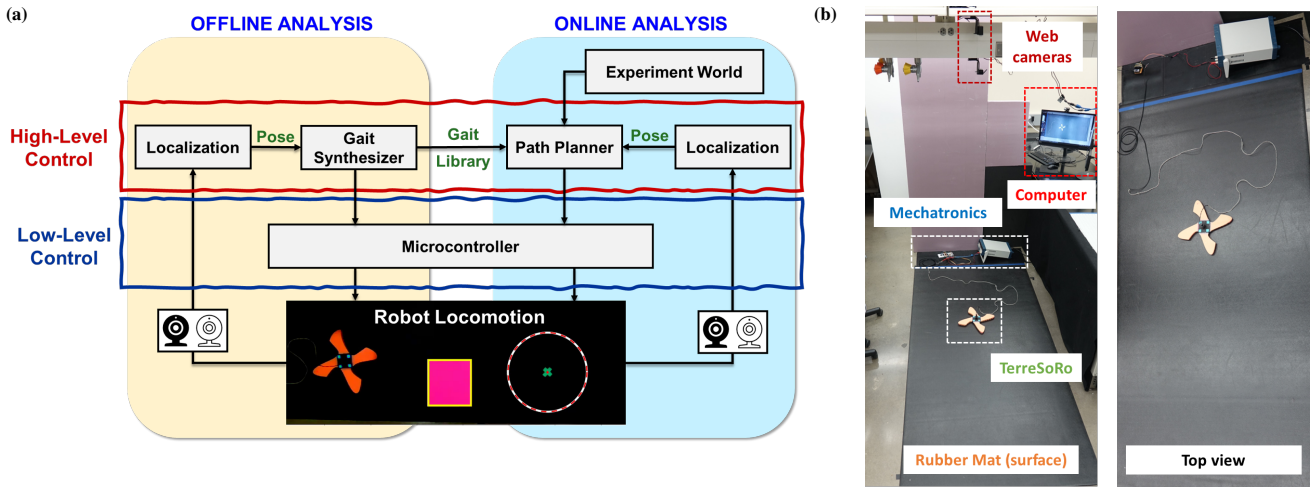


Fig. 3. (a) The control architecture combines offline analysis to generate the Gait Library for the online path planning with localization feedback. (b) The experimental setup involves system integration of high and low-level controllers involving processing in MATLAB and microcontroller.

for: (1) discovering useful gaits indigenous to a particular manufacturing instance, and (2) generating motion models that characterize these gaits for planning and control. Progression of these mobile robots toward autonomous field deployment entails an ability to both plan viable trajectories through an environment as well as accomplish some form of feedback-based control to track synthesized plans.

We employ a data-driven approach to synthesize gaits and predictive motion models for TerreSoRo; these inform a lattice-based planner whose gait sequence outputs, and accompanying locomotion trajectories, move TerreSoRo to prescribed goal locations within an obstacle-strewn environment. Feedback takes the form of trajectory re-planning when tracking error exceeds a pre-defined threshold.

III. METHODOLOGY, FRAMEWORK AND ALGORITHM

The control framework architecture comprises a high-level controller in MATLAB[®] that communicates with the low-level Arduino microprocessor to result in locomotion of the robot as summarized in Fig. 3. Localization is achieved by analyzing the webcam output and tracking the center of the robot. In the offline state, the Gait Synthesizer uses the motion data from the Euler cycle experiments to build the Gait Library. The gaits therein are then experimentally validated to store the expected motion data to feed into the path planner. In the online state, the experiment world (which maps the obstacles and initial robot pose) and the gait library are used to initialize the path planner. Real-time control is then achieved by comparing the expected pose from the path planner and the instantaneous experiment pose to inform the low-level controller and re-planning is performed as necessary.

A. Localization

The feedback to the robot controller plays a critical role in path re-planning. As can be seen in the experimental setup, Fig. 3b, two overhead webcams are used. One webcam is used to record HD video; the other is used for localization

and has its properties (e.g., contrast, brightness, etc.) adjusted to facilitate image segmentation of the four neon markers on the robot hub. Both webcam videos are processed in parallel using the MATLAB[®] Parallel Processing Toolbox[™]. The “localization core” video is processed through a image mask that highlights the markers located on the robot (blue markers on orange robot), obstacles (pink) and the target (green cross), as seen in Fig. 3. This video is stored and a pollable data queue accesses the data for localization at appropriate times. The pose estimation is performed using Arun’s method [27], which finds the least-squares solution to the pose by using singular value decomposition. Occlusion of the markers by the tether is managed by identifying the marker in the next time frame using the nearest neighbor. The occluded markers are reconstructed using the estimated pose. Camera capture occurs at $\sim 30\text{Hz}$. The robot pose estimation provides feedback at 6Hz on an Intel[®] Xeon[®] E5-1650 v4, as dictated by the computational complexity and the robot’s average motion profile.

B. Gait Library

Gaits are synthesized as described in Sec. II-B. For this research, five gaits are chosen - a rotation dominant gait R and a translation dominant gait T_1 with its permutations T_2 , T_3 , and T_4 . The limb actuation patterns are shown in Fig. 4a.

Each of the gaits are run for 120 gait cycles and the mean locomotion twist $\xi(p_i, \theta_i)$ is obtained that can be used by the path planner. The mean translation and rotation for each of these gaits is visually shown in Fig. 4. These plots highlight a key observation: the translation-dominant gaits T_1 , T_2 , T_3 , and T_4 are offset from each other by 90 degrees as expected (due to the limbs being positioned at 90 degree offsets). As the robot is fabricated to be rotationally symmetric, we anticipated that the twist magnitudes of these gaits would be identical; after all, they are the same gait with actuation initiated by a different limb. However, this is not true and highlights the sensitivity of soft robots to small manufacturing inaccuracies/non-uniformities. While control

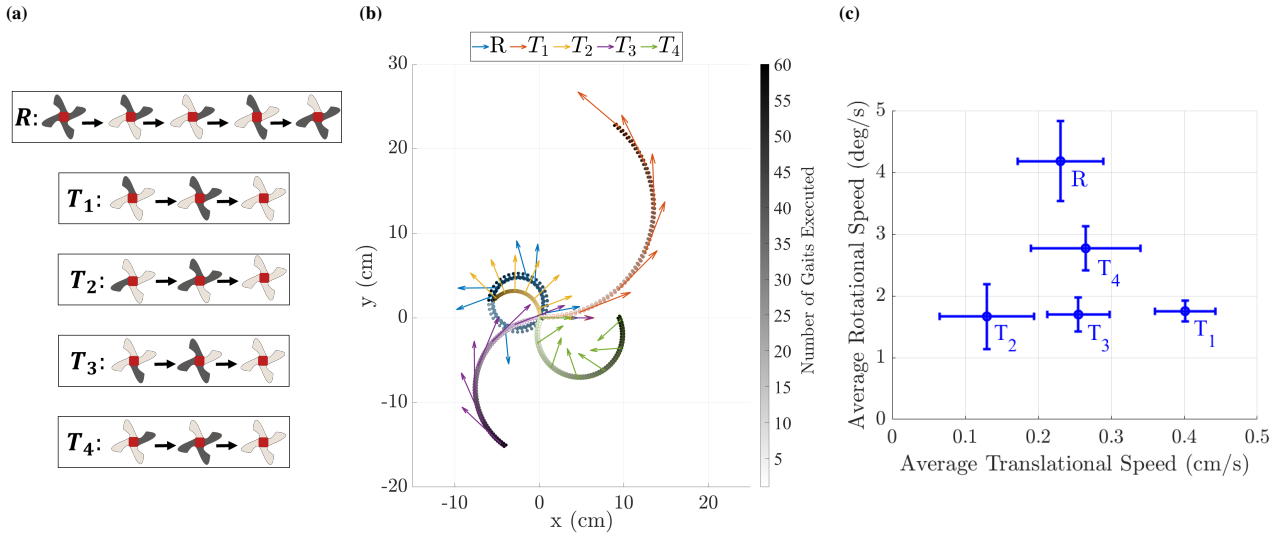


Fig. 4. (a) Gait Library comprising five gaits: rotation dominant gait R , and four translation dominant gaits $\{T_1, T_2, T_3, T_4\}$, where the last three gaits are permutations of the synthesized gait G . (b) Average locomotion of each gait executed for 60 cycles. Arrows indicate the orientation of the robot after every 10 cycles. (c) The average translation and rotation speed of each gait where the bars indicate the standard deviation.

parameters could potentially be adjusted to reduce these differences in behavior, it would be unlikely to eliminate them completely as the frictional effects appear to dominate. Moreover, soft robots are sensitive to small changes in the environment (e.g., small bumps in the substrate) as suggested by the error bars in Fig. 4b. The data-driven gait discovery and the path planning strategy accommodate these asymmetries and variations by treating each starting limb option as a distinct gait with individually (experimentally-) derived behaviors; feedback-based re-planning additionally serves to mitigate undesirable effects.

C. Trajectory Planning

Trajectory synthesis is computed on a binary image representation of an obstacle scenario. This input image is transcribed to a grid world cost map, $C(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$, where (x, y) denote grid coordinates and $C(x, y)$ denotes the cost associated with each grid location. Obstacle locations are characterized by high costs, which fall off radially with distance. A tunable parameter governs how quickly cost decays and is used to effectively dilate obstacles. Grid locations are classified as either *occupied* or *free* based on a pre-configured cost threshold. This results in a configuration space suitable for a point representation of the robot. Robot presence at any *occupied* grid location constitutes an obstacle collision. For a prescribed goal position $\mathbf{x}_{\text{goal}} \in E(2)$, cost-to-go $C_{\text{go}}(x, y; \mathbf{x}_{\text{goal}}) : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ across the grid world is quickly computed using an expanding wavefront approach. The cost-to-go allows us to discern the relative value of potential trajectory destinations within the world. Gait-based controlled trajectories are then designed within this grid world representation of the locomotion scenario.

Gait Models. Trajectory plans adhere to a rotate-then-translate motion paradigm. Synthesis entails searching for a sequence of these rotate-then-translate pairs, in order to move

the robot from a starting pose, $g_0 \in SE(2)$, to a desired goal location, $\mathbf{x}_{\text{goal}}$. This conceptually simplifies the trajectory planning problem; planning becomes an iterative process of: (1) ‘aiming’ (i.e. rotation), then (2) traveling in that direction (i.e. translation). This procedure terminates when the trajectory plan reaches a pre-defined radius δ_{goal} of the goal position $\mathbf{x}_{\text{goal}}$.

From the set of gaits that TerreSoRo is able to accomplish, we select a subset to be used in trajectory planning and control. A single gait is selected to accomplish rotationally-dominant motion. We denote this gait as R ; its behavior is characterized by a time-averaged body velocity twist $\xi^R \in \mathfrak{se}(2)$, and a gait periodicity of Q_R seconds. The remaining d gaits are characterized by translationally-dominant motion, and denoted as the set $\{T_1, T_2, \dots, T_d\}$. Each translational gait T_i exhibits a time-averaged body velocity twist $\xi^{T_i} \in \mathfrak{se}(2)$ and gait periodicity Q_{T_i} , where $i \in \{1 \dots d\}$.

Trajectory Synthesis. The planning strategy presented here employs a greedy breadth-first search through the space of possible TerreSoRo gait sequences. The approach successively expands sets of neighboring, collision-free trajectory destinations that may be reached by a single rotation-translation gait sequence. The neighboring destination with the smallest cost-to-go is then selected for subsequent expansion. Fig. 5 illustrates trajectory exploration and synthesis, through an obstacle-strewn environment. This strategy produced feasible trajectories, and the corresponding gait sequences needed to accomplish them, facilitating the robot’s intelligent traversal through a variety of obstacle arrangements.

Beginning at an initial expansion pose $g^{\text{expand}} \in SE(2)$, the planner first samples trajectory end points that may be reached by the rotational profile ξ^R , over durations of $n_R = 1 \dots N_R$ rotational gait periods. $N_R \in \mathbb{Z}^+$ is a fixed limit and computed such that $\xi_{\omega}^R \cdot N_R < \frac{\pi}{2}$. Reachable robot poses,

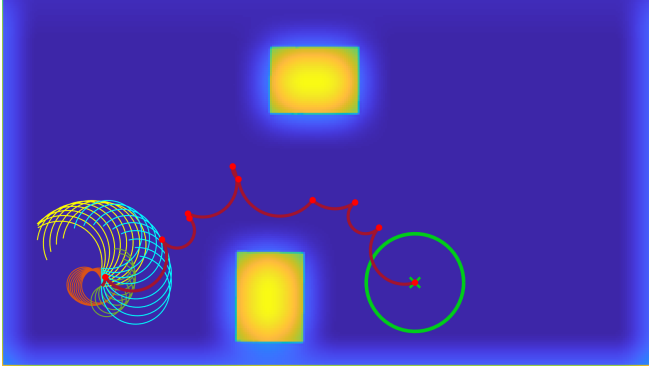


Fig. 5. Controlled trajectories are computed over a grid world cost map representation of the locomotion scenario, $C(x,y)$. Obstacle locations are assigned high costs (yellow) that decay with distance; locations far from obstacles entail low cost (dark blue). The trajectory solution (dark red) moves the robot from its starting pose g_0 (lower-left) to a prescribed goal position x_{goal} (green 'X'). An example trajectory search, beginning at the starting pose, is illustrated. Cyan, yellow, light green and orange trajectories depict explored expansions using each translational gait, T_1, T_2, T_3, T_4 , after initial rotation by gait R . The gait sequence leading to a subsequent pose with the least cost-to-go is selected; subsequent expansion then focuses on this new pose.

using the rotational gait R , are denoted $g_{n_R}^R \in SE(2)$ and expressed relative to the initial expansion pose g^{expand} . These poses are checked for collisions; if a collision is identified, the pose is discarded and removed from consideration going forward. Beginning from each collision-free $g_{n_R}^R$, trajectory end points are forward sampled for each translational motion profile ξ^{T_i} where $i = 1 \dots d$, and durations of $n_{T_i} = 1 \dots N_T$ translational gait periods; $N_T \in \mathbb{Z}^+$ is pre-configured and denotes the maximum number of consecutive cycles a translational gait may be run. The poses of these trajectory end points, relative to $g_{n_R}^R$, are denoted $g_{n_{T_i}}^{T_i}$. Their poses, relative to the initial expansion pose, are computed as $g_{n_R}^R \cdot g_{n_{T_i}}^{T_i}$; their corresponding spatial poses are $g(n_R, i, n_{T_i}) = g^{\text{expand}} \cdot g_{n_R}^R \cdot g_{n_{T_i}}^{T_i}$. The trajectory end points, as well as sub-sampled points along each trajectory segment, are tested for collisions; a trajectory expansion is discarded if a collision is detected. Cost-to-go C_{go} is evaluated at the (x,y) coordinates associated to each $g(n_R, i, n_{T_i}) \in SE(2)$. The motion sequence, described by $\{n_R, i, n_{T_i}\}$, and the corresponding trajectory destination $g(n_R, i, n_{T_i})$ that are associated with the lowest cost-to-go, are selected as the optimal control and subsequent node to be expanded, respectively. This procedure iterates until the selected pose $g(n_R, i, n_{T_i})$ falls within a threshold distance δ_{goal} of the goal position x_{goal} .

D. Path Recalculation

As observed, the locomotion gaits have both rotation and translation associated with them. As the robot performs gait cycles and switches gaits, the pose error does not always monotonically increase. Generally, the position error reaches some maximum value and then decreases. This observation is illustrated in Fig. 6. Consequently, path recalculation should be performed when the position error exceeds a user-defined error threshold upon completion of a gait sequence and/or at

user-defined intervals (i.e., every n gait cycles).

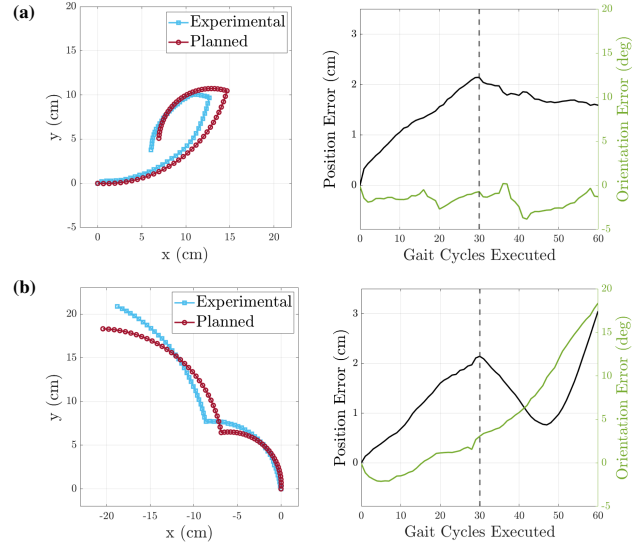


Fig. 6. The error between the path-planned and experimental poses of the robot for open-loop paths consisting of 60 gait cycles. The paths include a gait switch from (a) T_1 to T_2 and (b) T_2 to T_1 after 30 gait cycles (dashed vertical line). The plots indicate non-monotonic increase in the pose error.

IV. EXPERIMENTS AND DISCUSSION

To validate the closed-loop path planner, we performed experiments with three world scenarios with obstacles: (1) World 1 where the robot has to perform a ‘zig’ maneuver to go around the obstacle, (2) World 2 where the robot needs to perform a ‘zig’ and ‘zag’ motion to avoid obstacles and reach the goal, and (3) World 3 where the robot needs to go between two obstacles to reach the goal. All experiments are performed with the experimental setup and methodology discussed in Sec. III. A rubber garage mat is used as the substrate and paper cutouts are used for the obstacles. The TerreSoRo successfully maneuvers past the obstacles to reach the target position in all three scenarios. The results discussed in this section are visualized in the multimedia attachment to this paper.

World 1 consists of two obstacles where the robot is required to maneuver one obstacle to reach the target. The planner recalculates the path six times before it reaches the goal, as shown in Fig. 7a. The path is recalculated after the execution of a gait sequence if the position error exceeds a threshold. From the error plot, sudden drops in the error indicate path re-planning. World 2 requires the robot to perform two obstacle-avoiding maneuvers to reach the goal as shown in Fig. 7b. World 2 also triggers six path recalculations. In both World 1 and World 2, the position error both increases and decreases during the execution of gait sequences; it does not consistently monotonically increase before suddenly dropping at recalculation points. Thus, the robot is allowed to complete the current gait sequence, until the point of a gait switch. If the position error still exceeds the threshold, re-planning occurs.

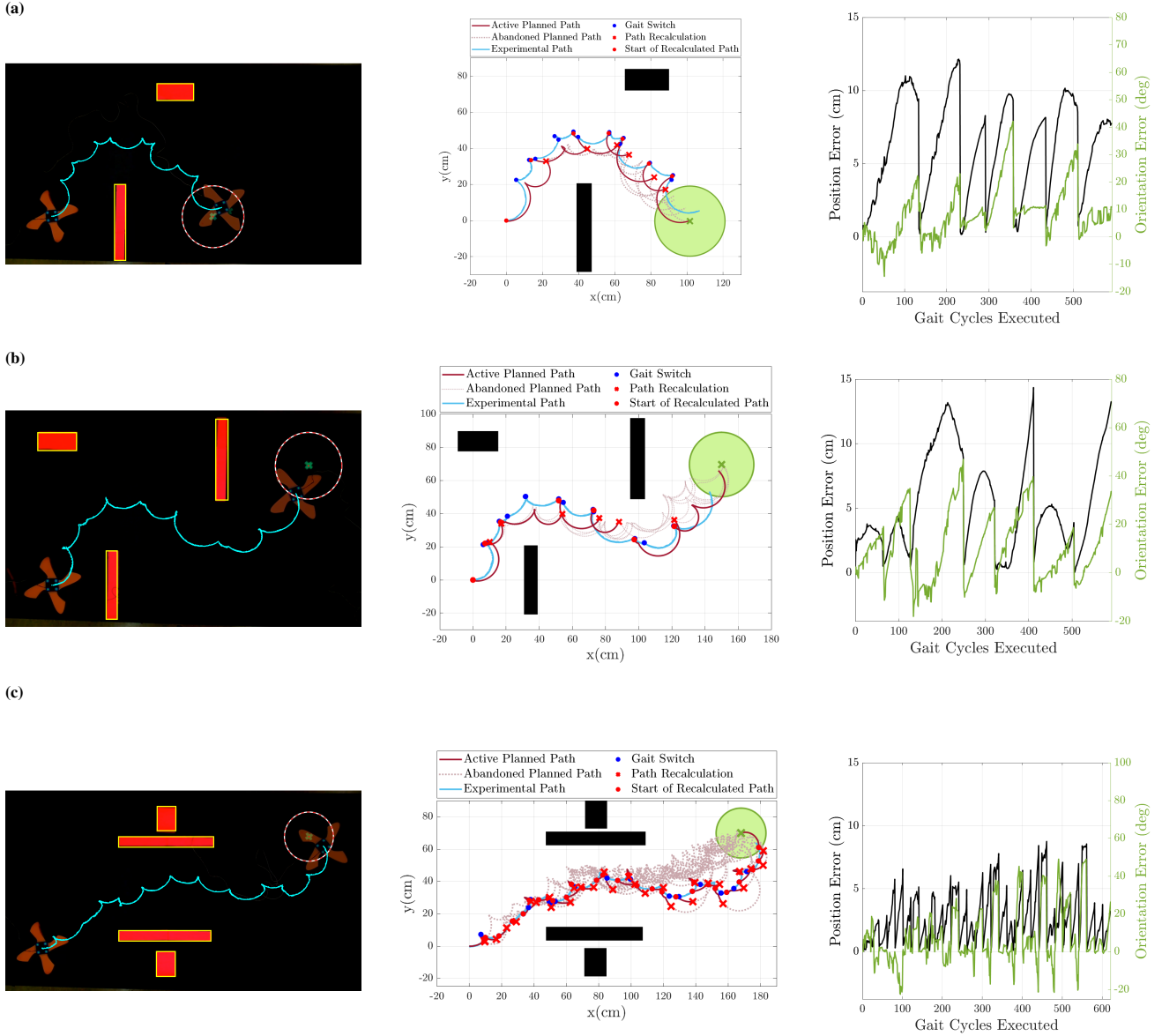


Fig. 7. Example implementations of the closed loop path planner for (a) World 1, (b) World 2, and (c) World 3. The images on the left show the starting and ending robot poses and its path (cyan) that avoids the obstacles (red) to reach the acceptance boundary (circle) around the target (green 'X'). The plots in the middle show the same experimental path of the robot (blue) compared to the planned path (maroon). The target is marked as a green 'X' with a green acceptance circle boundary around it. This plot also marks the moments of gait switches (blue dots), path recalculations (red), and planned paths (dotted pink) that are abandoned when re-planning occurs. The plots on the right show the position and orientation errors for these paths. Sudden corrections in the error indicate re-planning.

The World 3 scenario explores the ability of TerreSoRo to navigate narrow spaces, Fig. 7c. Here, to perform this delicate maneuver, the path re-planning is triggered after every 20 gait cycles. As observed, the position errors are smaller relative to previous world scenarios. However, this scenario also involves thirty recalculations and does not appear to exploit the potential decreases in the error that were seen in the previous worlds.

V. CONCLUSION AND FUTURE WORK

For the first time, this research presents successful closed-loop path planning and obstacle avoidance of the four-limb

motor-tendon actuated soft robot TerreSoRo. The experiment uses real-time visual feedback from low-cost webcams to perform localization, while the lattice-based path planner generates collision-free trajectories using a greedy breadth-first approach. As soft robots are more sensitive to factors like changes in the environment and manufacturing uncertainties, the locomotion gaits are synthesized using a data-driven environment-centric framework. Conceptually, this approach discretizes the factors dominating the environment-robot interaction and synthesizes the tracked motion of these interactions to find translation and rotation gaits. The path planner generates sequences of gaits that are pairs of

rotation-then-translation. The synthesized gaits for the TerreSoRo have coupled translation and rotation. Consequently, the path is recalculated when the position error exceeds a threshold after completion of a gait sequence, or at a user-defined interval. The framework is validated on complex world scenarios with obstacles that require the robot to perform challenging maneuvers. The non-uniform nature of the surface/environment and the intentional slipping action of the robot limbs further complicate this challenge.

Future work of this research involves extending the framework to a larger gait library that allows for locomotion on different surfaces. Furthermore, work will be done to adapt the path planner to incorporate the probabilistic nature of locomotion gaits.

REFERENCES

- [1] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Science robotics*, vol. 1, no. 1, p. eaah3690, 2016.
- [2] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [3] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Transactions on Robotics*, 2023.
- [4] Y. Sun, A. Abudula, H. Yang, S.-S. Chiang, Z. Wan, S. Ozel, R. Hall, E. Skorina, M. Luo, and C. D. Onal, "Soft mobile robots: A review of soft robotic locomotion modes," *Current Robotics Reports*, vol. 2, pp. 371–397, 2021.
- [5] A. H. Chang and P. A. Vela, "Shape-centric modeling for control of traveling wave rectilinear locomotion on snake-like robots," *Robotics and Autonomous Systems*, vol. 124, p. 103406, 2020.
- [6] G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 781–793, 1995.
- [7] A. H. Chang, C. Freeman, A. N. Mahendran, V. Vikas, and P. A. Vela, "Shape-centric modeling for soft robot inchworm locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 645–652.
- [8] E. Coevoet, A. Escande, and C. Duriez, "Soft robots locomotion and manipulation control using fem simulation and quadratic programming," in *IEEE International Conference on Soft Robotics*, 2019, pp. 739–745.
- [9] J. M. Bern, P. Banzet, R. Poranne, and S. Coros, "Trajectory optimization for cable-driven soft robot locomotion," in *Robotics: Science and Systems*, vol. 1, no. 3, 2019.
- [10] S. Mao, E. Dong, H. Jin, M. Xu, S. Zhang, J. Yang, and K. H. Low, "Gait study and pattern generation of a starfish-like soft robot with flexible rays actuated by smas," *Journal of Bionic Engineering*, vol. 11, no. 3, pp. 400–411, 2014.
- [11] V. Vikas, P. Grover, and B. Trimmer, "Model-free control framework for multi-limb soft robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 1111–1116.
- [12] C. Freeman, A. N. Mahendran, and V. Vikas, "Gait synthesis of limbed terrestrial soft robots via environment-centric learning and optimization," (under review) 2023.
- [13] Z. J. Patterson, A. P. Sabelhaus, K. Chin, T. Hellebrekers, and C. Majidi, "An untethered brittle star-inspired soft robot for closed-loop underwater locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 8758–8764.
- [14] Z. Liu and K. Karydis, "Position control and variable-height trajectory tracking of a soft pneumatic legged robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 1708–1709.
- [15] S. Hamill, J. Whitehead, P. Ferenz, R. F. Shepherd, and H. Kress-Gazit, "Resilient task planning and execution for reactive soft robots," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 5148–5154.
- [16] M. Luo, Z. Wan, Y. Sun, E. H. Skorina, W. Tao, F. Chen, L. Gopalka, H. Yang, and C. D. Onal, "Motion planning and iterative learning control of a modular soft robotic snake," *Frontiers in Robotics and AI*, vol. 7, p. 599242, 2020.
- [17] A. Dornbush, K. Vijayakumar, S. Bardapurkar, F. Islam, M. Ito, and M. Likhachev, "A single-planner approach to multi-modal humanoid mobility," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 4334–4341.
- [18] J. Barry, L. P. Kaelbling, and T. Lozano-Pérez, "A hierarchical approach to manipulation with diverse actions," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1799–1806.
- [19] K. Hauser, V. Ng-Thow-Hing, and H. Gonzalez-Baños, "Multi-modal motion planning for a humanoid robot manipulation task," in *The International Symposium of Robotics Research*. Springer, 2011, pp. 307–317.
- [20] H. T. Suh, X. Xiong, A. Singletary, A. D. Ames, and J. W. Burdick, "Energy-efficient motion planning for multi-modal hybrid locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 7027–7033.
- [21] K. Kon, M. Tanaka, and K. Tanaka, "Mixed integer programming-based semiautonomous step climbing of a snake robot considering sensing strategy," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 252–264, 2015.
- [22] C. Freeman, M. Maynard, and V. Vikas, "Topology and morphology design of spherically reconfigurable homogeneous modular soft robots," *Soft Robotics*, 2022.
- [23] C. Freeman, J. Conzola, and V. Vikas, "Topology Design and Optimization of Modular Soft Robots Capable of Homogenous and Heterogenous Reconfiguration," *Journal of Computational and Nonlinear Dynamics*, vol. 18, no. 6, p. 061007, 04 2023.
- [24] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 2013, vol. 184.
- [25] V. Radhakrishnan, "Locomotion: Dealing with friction," *Proceedings of the National Academy of Sciences*, vol. 95, no. 10, pp. 5448–5455, May 1998.
- [26] H. C. Astley, "Getting around when you're round: quantitative analysis of the locomotion of the blunt-spined brittle star, ophiocoma echinata," *Journal of Experimental Biology*, vol. 215, no. 11, pp. 1923–1929, 2012.
- [27] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 698–700, 1987.