2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Feasible Rolling Trajectory Generation and Control for Tensegrity Robots

Songyuan Liu, Qingkai Yang, *Senior Member, IEEE*, Zichen Tao, Yun Gui, Jiaxu Shi, Hao Fang, *Member, IEEE*

***Abstract*— Due to the multi-node and multi-contact motion characteristics of tensegrity robots, existing methods fail to generate feasible reference rolling trajectories, and controllers are also limited to open-loop approaches. To address this issue, we utilize motion decomposition to extract the motion phase that should be the primary focus. Subsequently, we propose a method combining form-finding-based critical configuration search and polynomial trajectories to generate feasible trajectories. Then, an iLQR controller that accounts for reducing actuator load is designed for trajectory tracking control. A key distinction from existing methods is that our approach eliminates the need for reset operations after each rolling cycle. The results of simulations and physical experiments demonstrate that the robot achieves continuous rolling, with improvements of 18.3% in speed and 34.4% in actuation load compared to existing works.**

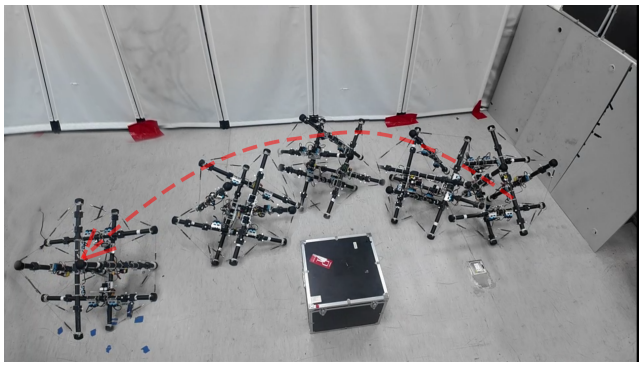

Fig. 1. Continuous rolling motion of the six-bar tensegrity robots. The robot advances toward the target position by sequentially switching its contact polygons, a behavior termed continuous rolling.

I. INTRODUCTION

Tensegrity robots, characterized by their unique combination of rigid bars and flexible cables, have emerged as a promising paradigm in robotics due to their inherent compliance, lightweight design, and adaptability to complex environments [1]-[3]. These robots, inspired by the principles of tensegrity structures, exhibit remarkable properties such as passive compliance, variable stiffness, and intrinsic safety, making them particularly suitable for applications in unstructured terrains, disaster response [4], [5].

Among the various forms of motion achievable by tensegrity robots, rolling locomotion stands out as a highly efficient and versatile mode of mobility, enabling the robot to traverse uneven surfaces and navigate through confined spaces [6], [7]. However, despite their potential, the control of tensegrity robots, particularly in achieving continuous and stable rolling locomotion, remains a significant challenge due to their complex dynamics, high-dimensional state space, and the intricate interplay between internal forces and external environmental interactions [8]-[10]. The high degree of freedom and the nonlinear dynamics of tensegrity robots make it difficult to predict and control their behavior accurately. Additionally, the interaction between the robot and its environment, particularly during rolling locomotion, adds another layer of complexity, as the robot must continuously adapt its configuration to maintain stability and achieve desired motion.

The rolling locomotion of tensegrity robots has been a focal point of research due to its potential for efficient mobility in unstructured environments. Early approaches to rolling control relied on open-loop strategies, such as hand-tuned stepwise methods [11], and rule-based control schemes [12], [13]. In [14] and [15], several novel control algorithms and actuation schemes were also investigated. While these methods demonstrated the feasibility of generating rolling gaits, they often lacked adaptability to varying configurations and environmental conditions, leading to unpredictable behavior and limited control precision.

More advanced techniques, such as central pattern generators [16], evolutionary algorithms [17], and deep reinforcement learning [18]-[21], have been employed to achieve continuous rolling locomotion. These methods have shown promise in generating complex rolling gaits and adapting to dynamic environments. However, they often require extensive manual tuning or large amounts of training data, which can be computationally expensive and may not generalize well to real-world scenarios. Moreover, many of these approaches operate independently of the robot's dynamic model, which limits their ability to achieve precise and stable control.

One of the key challenges in rolling control is the complexity of its trajectory and the associated control difficulties.

*This work was supported in part by the National Key Research and Development Program of China under No. 2022YFB4702000, in part by the NSFC under Grants 62373048, 62133002, 62088101.

S. Liu, Q. Yang, Z. Tao, Y. Gui, J. Shi, H. Fang are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, and also with National Key Laboratory of Autonomous Intelligent Unmanned Systems, Beijing 100081, China (e-mail: 3120205435@bit.edu.cn; qingkai.yang@bit.edu.cn; zichentao@outlook.com; 3220245213@bit.edu.cn; 3220241261@bit.edu.cn; fangh@bit.edu.cn).

The high-dimensional state of the tensegrity robot, combined with its interaction with the ground, makes the complete description and tracking control of its trajectory an almost unattainable goal. Additionally, the transition between rolling gaits, which is essential for continuous rolling locomotion, remains an under-explored area. In this paper, we address the aforementioned challenges by proposing a novel approach to continuous rolling control for six-bar tensegrity robots. The proposed approach enables the robot to perform multiple consecutive rolls without the need for resetting, significantly enhancing motion continuity and efficiency. The effectiveness of the proposed approach has been validated through both simulations and physical prototype experiments. The key contributions of this work are as follows:

Motion decomposition: By distinguishing between different driving modes, the continuous rolling motion is decomposed into a three-point contact phase (stance phase) and a two-point contact phase (rolling phase). This decomposition effectively reduces the complexity of motion description.

Trajectory generation: To obtain the trajectory for each stance phase (the reason why the trajectory generation and control of the rolling phase are not considered will be explained in subsequent sections), we propose an autonomous continuous motion trajectory generation method that integrates motion switching and critical configuration search for tensegrity robots.

Control and experimental validation: For the stance phase, we design an iLQR controller that ensures the transition from each stance phase to the rolling phase while considering the reduction of actuator load. The proposed controller is validated through simulation and physical experiments.

The remainder of this paper is organized as follows: Section II introduces the motion decomposition approach and the corresponding dynamic model for the tensegrity robots. Section III details the proposed continuous rolling trajectory generation and control framework. Section IV presents the simulation and experimental results, followed by a discussion of the findings. Finally, Section V concludes the paper and outlines future research directions.

II. Motion Decomposition for Tensegrity Robots

This section establishes the theoretical foundation for continuous rolling control. We introduce the six-bar tensegrity robot's structural properties and hybrid dynamics model, focusing on the motion decomposition scheme that separates locomotion into deformation and rolling phases.

A. The six-bar tensegrity robot

Compared with other tensegrity robots, the six-bar tensegrity robot has a shape close to a sphere and can generate continuous rolling motion. When compared to more sphere-like tensegrity robots (e.g., the twelve-bar robot), it features a simpler structure, avoids the issue of an excessive number of actuators, and is thus more suitable for verifying the method proposed in this paper. The six-bar tensegrity robot comprises 6 rigid bars and 24 elastic cables interconnected in a symmetric truncated octahedron geometry. The rigid

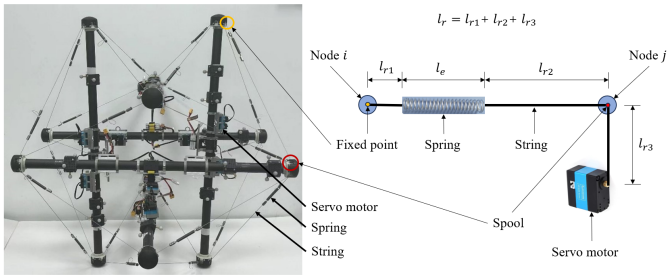

Fig. 2. The six-bar tensegrity robot. The bars are hollow carbon fiber tubes. The elastic cables consist of steel strings connected with steel springs. Other connecting mechanisms (such as spools, servo motor fixing devices, and battery mounts) are manufactured via CNC (Computer Numerical Control) machining. The marker mounting modules of the motion capture system are fabricated via 3D printing and fixed to both sides of the bars using hot-melt adhesive. The yellow circles enclose fixed points, which use a fixed connectionthere is no relative displacement between the cables and the nodes. The red circles enclose spools, the cables can slide around the spools under the drive of the servo motors.

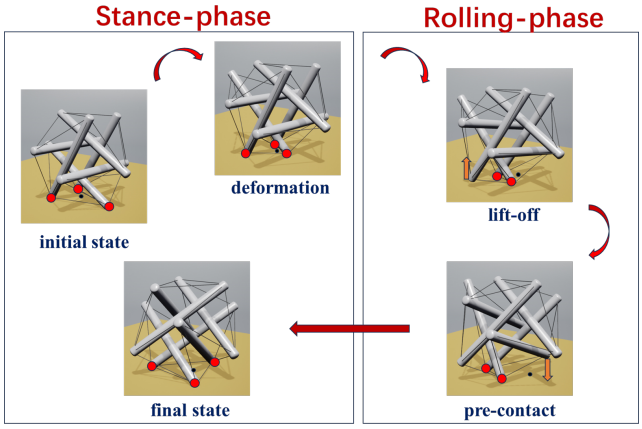

Fig. 3. Motion decomposition of a six-bar tensegrity robot. The red dots in the diagram represent contact points, the black dots represent the projections of the center of mass onto the ground at the corresponding moments.

bars, serving as compressive elements, are joined at 12 nodes through spherical joints, forming a spatial framework, while the elastic cables constitute a continuous tensile network maintained by pretension.

The robot's locomotion is achieved via a fully actuated cable-driven system (Fig. 2): Each cable assembly comprises two rigid segments and a spring in series, connected to a servo motor with position feedback. This configuration allows centralized controllers to adjust cable lengths dynamically, thereby modulating the robot's structural morphology and shifting its center of mass (CoM) for locomotion. When specific cables are motor-driven to contract, localized tension variations induce global structural tilting, displacing the CoM projection beyond the current contact face boundary, thereby triggering rolling motion.

B. Motion decomposition

The six-bar tensegrity robot exhibits no kinematic constraints between nodes except those shared by two nodes

on the same bar, resulting in a total of 36 degrees of freedom (DoFs). Describing its motion (particularly continuous rolling) in such a high-dimensional state space is considered complex and intractable. Inspired by phase-divided motion planning strategies in legged robot research [22], [23], we propose an analogous motion decomposition framework for the tensegrity robot.

Motion decomposition: understanding the locomotion pattern of tensegrity robots is essential for achieving controlled rolling behaviors. The motion cycle of a tensegrity robot can be decomposed into two primary phases based on ground contact configurations:

- **Stance phase:** The robot maintains three contact points with the ground, forming a stable triangular contact face. During this phase, the structure remains in static equilibrium, enabling energy-efficient posture adjustments and preparatory deformations for subsequent rolling.
- **Rolling phase:** Characterized by two contact points, the robot becomes dynamically unstable and undergoes torque-induced rotation. This process is mainly driven by gravity, featuring a short duration, rapid state changes, and limited influence from active control.

Fig. 3 provides a refined breakdown of the rolling process: the robot starts from the initial state, undergoes deformation driven by the actuators, the lift-off of one contact point marks the beginning of the roll phase, and the pre-contact of a new contact point indicates that the robot is about to enter the next stance phase.

For the tensegrity robot, the rolling phase is a passive and rapid process under the influence of gravity. In contrast, the deformation phase requires precise control to ensure successful lift-off. Therefore, this paper focuses on this particular phase. The significance of motion decomposition lies in decoupling the complex global motion into two distinct phases. This approach allows us to achieve continuous rolling by focusing solely on the deformation control process, significantly reducing the complexity of motion description while retaining essential dynamic characteristics.

C. Hybrid system of the six-Bar tensegrity robot

Through motion decomposition, the locomotion of the tensegrity robot is regarded as the switching between the two phases. Therefore, we consider constructing a corresponding hybrid dynamical model to simulate and control its motion. The locomotion of the six-bar tensegrity robot is inherently governed by a hybrid dynamical system due to its alternating contact states. Formally, the hybrid system is defined as:

$$\mathscr{H} = (\mathscr{D}, \mathscr{S}, \Delta, \mathscr{U}). \tag{1}$$

where:

- Domains $\mathscr{D} = \{D_{\text{stance}}, D_{\text{rolling}}\}$:
  - D_{stance}: Three-node contact with static equilibrium
  - D_{rolling}: Two-node contact with rotational dynamics
- Guards $\mathscr{S} = \{S_{\text{stance}\to\text{rolling}}, S_{\text{rolling}\to\text{stance}}\}$:
  - $S_{\text{stance}\to\text{rolling}}$: CoM projection crosses contact polygon boundary
  - $S_{\text{rolling}\to\text{stance}}$: New contact establishment detected
- Reset maps Δ: During contact transitions, varying interaction forces necessitate updates to the system state, governed by the Hertz contact model.
- Controls $\mathscr{U}$: Cable length variations induced by servo motor actuation $u \in \mathbb{R}^{24}$.

Continuous dynamics: In [24], a dynamic model has been established to compute the state of the six-bar tensegrity robot under the influence of driving forces and environmental factors. The dynamics equation for the tensegrity robot is presented as

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + F(\theta) + U_g(\theta) + Gu = 0 \tag{2}$$

where $\theta \in \mathbb{R}^{36}$ denotes the state vector composed of nodal coordinates. The inertia matrix $M(\theta) \in \mathbb{R}^{36\times 36}$ characterizes the mass distribution of the structure, while $C(\theta,\dot{\theta}) \in \mathbb{R}^{36\times 36}$ encapsulates Coriolis and centrifugal forces. The term $F(\theta) \in \mathbb{R}^{36}$ represents dissipative forces including structural damping and environmental friction. The gravitational potential gradient $U_g(\theta) \in \mathbb{R}^{36}$ captures gravitational effects, and $G \in \mathbb{R}^{36\times 24}$ denotes the actuation mapping matrix. When the guard conditions specified in the hybrid system formulation are met, the dynamics computation automatically switches to the continuous dynamics associated with the target phase, thereby enabling phase-specific state evolution.

III. METHODOLOGY

Building on Section II, this section presents the core methodology for continuous rolling control, integrating form-finding based trajectory generation, an iLQR controller for hybrid dynamics, and motion decomposition logic.

A. Trajectory generation

To achieve the rolling of a tensegrity robot, it is necessary to control each node of the robot to reach preset coordinates during the deformation process. For the sake of simplifying the description, we refer to the set of nodal coordinates of the tensegrity structure as its configuration; among these configurations, the one that enables the robot to enter the rolling phase is called the critical configuration (i.e., the set of preset target nodal coordinates). This paper proposes a form-finding-based critical configuration search method. Form-finding is a category of methods for determining the unknown configuration of a tensegrity structure, which can calculate the base coordinates describing the configuration based on the structural topology. Herein, we reference the conclusions from [25] and [26]:

For a 3-d structure, to ensure non-degenerate configurations, the force density matrix D must satisfy the rank-deficiency condition:

$$\text{rank}(D) \le n - (d+1). \tag{3}$$

Particle Swarm Optimization (PSO) is employed to solve for force densities q, with the fitness function minimizing the sum of the first $d+1$ eigenvalues of D:

$$f = \sum_{i=1}^{d+1} |\lambda_i|, \tag{4}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are eigenvalues of D. The eigenvectors corresponding to the $d+1$ zero eigenvalues of D form the base coordinate matrix $V = [v_1, v_2, \cdots, v_{d+1}]$.

To obtain the configuration that enables the robot to roll, we introduce the robots rolling condition, which requires the CoM projection to align with the rolling axis. For a contact face formed by nodes i, j, k and rolling axis $\overrightarrow{jk}$, the CoM coordinate (x_g, y_g) is computed as

$$\begin{cases} x_g = \frac{1}{n}\sum_{m=1}^{n} x_m \\ y_g = \frac{1}{n}\sum_{m=1}^{n} y_m \end{cases}, \tag{5}$$

where $x, y, z \in \mathbb{R}^n$ are the Cartesian coordinate vectors of nodes in x, y, z directions. The rolling condition is mathematically expressed as:

$$g_c = (x_g - x_j)(y_k - y_j) - (y_g - y_j)(x_k - x_j) = 0. \tag{6}$$

Nodal coordinates satisfying force equilibrium can be represented as linear combinations of base eigenvectors:

$$\begin{cases} x = Vw_1 \\ y = Vw_2 \\ z = Vw_3 \end{cases}, \tag{7}$$

where $w_1, w_2, w_3 \in \mathbb{R}^{d+1}$ are linear combination coefficients. Substituting the coordinate expressions into the rolling condition, we solve for the coefficient matrix $W = [w_1, w_2, w_3]^T \in \mathbb{R}^{(d+1)\times 3}$ via the linear constraint:

$$S \begin{bmatrix} Vw_1 \\ Vw_2 \\ Vw_3 \end{bmatrix} = \begin{bmatrix} x_g \\ y_g \end{bmatrix}, \tag{8}$$

where $S = \frac{1}{n}\begin{bmatrix} 1_{1\times n} & 0_{1\times n} & 0_{1\times n} \\ 0_{1\times n} & 1_{1\times n} & 0_{1\times n} \end{bmatrix}$ is the CoM coefficient matrix. The critical configuration nodal coordinate matrix is finally obtained as:

$$\theta_d = VW. \tag{9}$$

To reduce actuation load, we minimize robot deformation by optimizing the CoM position parameter $p \in (0,1)$ (defining CoM's relative position on the rolling axis: $x_g = x_k + p(x_j - x_k), y_g = y_k + p(y_j - y_k)$). The optimization problem is formulated as

$$\begin{cases} \min\limits_{p} & g_d = \sum_{k=1}^{24} |d_k - d_0| \\ \text{s.t.} & \begin{cases} g_c = 0 \\ z_i = z_j = z_k = 0 \end{cases} \end{cases}, \tag{10}$$

where d_k is the length of the k-th cable, and d_0 is the cable length in the standard symmetric configuration. The optimization problem is solved by PSO method.

Given the initial configuration θ_0 and target critical configuration θ_d, we generate smooth transition trajectories through constrained quintic polynomial interpolation:

$$\theta(t) = \sum_{k=0}^{5} a_k \left(\frac{t}{T}\right)^k, \quad t \in [0, T]. \tag{11}$$

Boundary Conditions:

$$\theta(0) = \theta_0, \ \dot{\theta}(0) = 0, \ \ddot{\theta}(0) = 0, \tag{12}$$

$$\theta(T) = \theta_d, \ \dot{\theta}(T) = 0, \ \ddot{\theta}(T) = 0. \tag{13}$$

As exemplified in Fig. 4, the reference trajectory guides all 12 nodes from the initial configuration θ_0 to the target critical configuration θ_d through constrained polynomial interpolation. It is worth noting that the transition from the initial configuration to the critical configuration represents a specific case of the stance phase. In other words, by controlling the robot to reach the critical configuration, we can trigger its entry into the rolling phase. The design of the controller will be detailed in the next subsection.

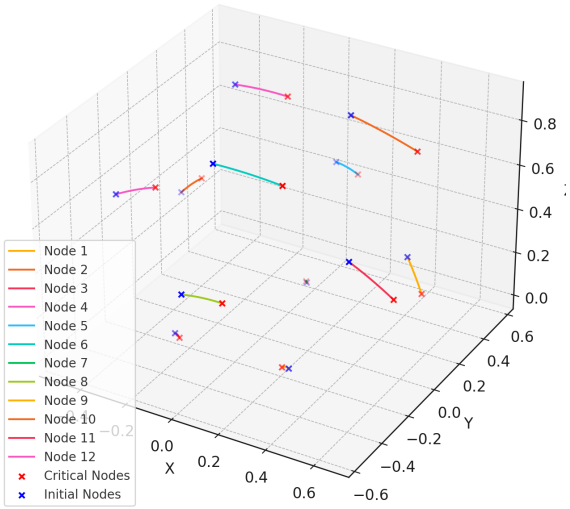

Fig. 4. Nodal trajectories. The initial nodes represent the robot's current configuration, while the critical nodes denote the configuration immediately before entering the rolling phase.

B. iLQR controller for stance phase

The stance phase dynamics are governed by the hybrid system defined in Section II. For iLQR implementation, we adopt the following state-space representation:

$$\mathbf{x} = [\theta^T, \dot{\theta}^T]^T \in \mathbb{R}^{72}, \quad \mathbf{u} = \Delta\mathbf{l} \in \mathbb{R}^{24}, \tag{14}$$

where $\theta \in \mathbb{R}^{36}$ denotes nodal coordinates, $\dot{\theta}$ their velocities, and $\Delta\mathbf{l}$ the cable length variations. The discrete-time dynamics are linearized as:

$$\delta\mathbf{x}_{k+1} = \mathbf{A}_k \delta\mathbf{x}_k + \mathbf{B}_k \delta\mathbf{u}_k \tag{15}$$

with Jacobians $\mathbf{A}_k = \frac{\partial f}{\partial \mathbf{x}}|_{\mathbf{x}_k, \mathbf{u}_k}$ and $\mathbf{B}_k = \frac{\partial f}{\partial \mathbf{u}}|_{\mathbf{x}_k, \mathbf{u}_k}$ computed from the continuous dynamics (Eq.2). The cost function is

given as

$$J = \sum_{k=0}^{N-1} \ell(\mathbf{x}_k, \mathbf{u}_k) + \ell_f(\mathbf{x}_N), \tag{16}$$

$$\ell(\mathbf{x}, \mathbf{u}) = \underbrace{\|\mathbf{x} - \mathbf{x}^{\text{ref}}\|^2_{\mathbf{Q}}}_{\text{Tracking}} + \underbrace{\|\mathbf{u}\|^2_{\mathbf{R}}}_{\text{Effort}} + \underbrace{\lambda \|\nabla V_e(\theta)\|^2}_{\text{Elasticity}}, \tag{17}$$

$$\ell_f(\mathbf{x}) = \underbrace{\|\mathbf{x} - \mathbf{x}^{\text{ref}}\|^2_{\mathbf{Q}_f}}_{\text{Terminal cost}}. \tag{18}$$

The weighting matrix $\mathbf{Q} \in \mathbb{R}^{72\times72}$ applies position tracking weights 10^3 to node coordinates and 10^2 to velocities through its block-diagonal structure. The control penalty matrix $\mathbf{R} \in \mathbb{R}^{24\times24}$ uses uniform weights 10^{-2} to regularize cable length changes. The elastic potential gradient term $\lambda\|\nabla V_e\|^2$ with $\lambda = 0.1$ penalizes excessive structural deformations by evaluating the spatial derivative of elastic energy V_e computed from nodal displacements. This multi-objective formulation ensures precise nodal trajectory tracking while maintaining minimal control effort and structural integrity through energy-aware regularization.

Algorithm 1 Interitive LQR Controller for Tensegrity Robots

Require: Initial state $\mathbf{x}_0$, reference trajectory $\{\mathbf{x}_k^{\text{ref}}\}_{k=0}^{N}$, max iterations M

Ensure: Optimal control sequence $\{\mathbf{u}_k^*\}_{k=0}^{N-1}$

Initialize control sequence $\{\mathbf{u}_k^{(0)}\}$
Initialize cost $J^{(0)} \leftarrow \infty$
for $m \leftarrow 1$ **to** M **do**
  $\{\mathbf{x}_k^{(m)}\} \leftarrow \text{ForwardSim}(\mathbf{x}_0, \{\mathbf{u}_k^{(m-1)}\})$ ▷ Nonlinear dynamics rollout
  $J^{(m)} \leftarrow \sum_{k=0}^{N} \ell(\mathbf{x}_k^{(m)}, \mathbf{u}_k^{(m-1)})$ ▷ Evaluate Eq. 16
  Compute gains $\mathbf{K}_k$, $\mathbf{k}_k$ via BackwardPass($\{\mathbf{x}_k^{(m)}\}$) ▷ Riccati solver
  Update controls: $\mathbf{u}_k^{(m)} \leftarrow \mathbf{u}_k^{(m-1)} + \mathbf{K}_k(\mathbf{x}_k^{(m)} - \mathbf{x}_k^{\text{ref}}) + \mathbf{k}_k$
  if $|J^{(m)} - J^{(m-1)}| < \varepsilon$ **then**
    break
  end if
end for

The iLQR algorithm implements deformation control for tensegrity robots through an iterative forward-backward optimization framework. Forward pass simulates nonlinear dynamics to generate state trajectories while evaluating three cost components: tracking error ($\delta\mathbf{x}_k$), control effort ($\mathbf{u}^T\mathbf{R}\mathbf{u}$), and elastic potential gradient ($\|\nabla V_e\|^2$). Backward pass solves Riccati equations using system Jacobians ($\mathbf{A}_k, \mathbf{B}_k$) and potential energy Hessian ($\nabla^2 V_e$), deriving optimal feedback gains $\mathbf{K}_k$ and feedforward corrections $\mathbf{k}_k$. Control update applies these gains to trajectory deviations, progressively optimizing control inputs until convergence ($|J^{(m)} - J^{(m-1)}| < \varepsilon$).

C. Control framework

As illustrated in Fig. 5, the tensegrity robot achieves continuous rolling through the following coordinated steps:

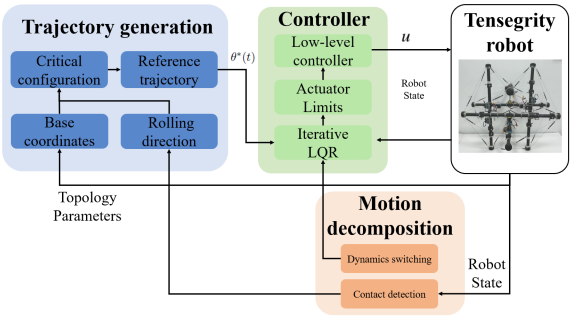

Fig. 5. Control diagram.

1) **Motion decomposition**: Based on the current contact information, the motion decomposition (orange module) determines the robot's current phase, switches the dynamic model accordingly, and obtains the desired rolling direction from the preset rolling sequence, which is then provided to the trajectory generator.
2) **Trajectory generation**: According to the subsequent desired rolling direction, the rolling axis is selected, the critical configuration that satisfies the rolling conditions is calculated, and a polynomial trajectory is generated to serve as the reference for the iLQR controller.
3) **Deformation control**: The iLQR controller computes the feedback matrix and control inputs through the forward pass and backward pass. After amplitude limiting, these control inputs are sent to the low-level controller.
4) **Physical execution**: The tensegrity robot executes the control commands while providing state feedback. This closed-loop process enables continuous contact transition and energy-efficient rolling.

IV. SIMULATION AND PHYSICAL EXPERIMENTS

In this section, we discuss the implementation of our proposed control framework, investigate some of its properties through simulation results, and validate the approach in real-world experiments.

A. Simulation

1) Simulation setup: Simulations is carried out on the matlab software, the differential equations of six-bar tensegrity robot dynamic are solved by Runge-Kutta method. The parameters applied in the dynamic model are shown in Table I, which are determined experimentally. To evaluate the effectiveness of the proposed controller in generating continuous rolling motion, we predefined a sequence of ground contact transitions, enabling the robot to move along an approximately straight path. The controller's objective is to drive the robot from each stance phase to the predefined rolling phase.

2) Simulation results: Fig. 6 demonstrates the rolling performance of the robot in simulation. The results show five consecutive rolling motions, each aligned with the predefined rolling direction, indicating that the iLQR controller

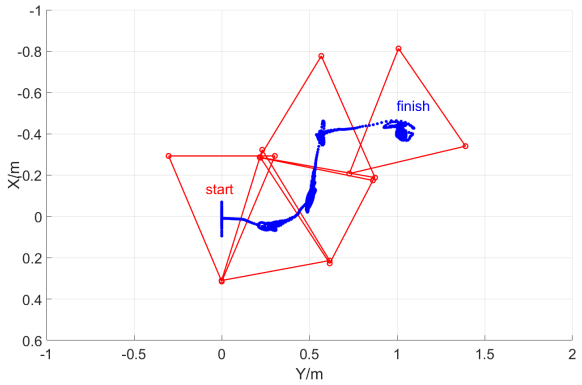

Fig. 6. Trajectory of the robot's center of mass (GCoM) during the rolling process in simulation. The robot alternates between stance phase and rolling phase (Top view).

successfully drives the robot to the critical state and triggers rolling during every deformation phase. The penultimate and last triangular faces appear to be relatively far apart, in fact, there is a narrower contact face between them. The center of mass (CoM) easily escapes this narrow contact face, causing the robot to not stay on it and instead continue rolling to the next contact face. In terms of motion efficiency, our method achieves an average CoM velocity of 45.2 cm/s, outperforming the 38.2 cm/s reported in [27] under the same maximum motor speed. This improvement stems from fundamental differences in control strategies: while [27] employs Model Predictive Control (MPC) to optimize nodal velocity directions (forcing all nodes to move in the predefined direction), our approach better captures the intrinsic rolling dynamics of tensegrity robots. The MPC-based method results in inchworm-like locomotion rather than true rolling motion, explaining its lower efficiency. The simulation environment provides a clear visualization of the robot's trajectory and rolling process, with color coding potentially used to distinguish different components or motion states.

TABLE I

SIMULATION PARAMETERS FOR THE SIX-BAR TENSEGRITY ROBOT

Parameter	Value
Bar length (l_b)	0.9 m
Bar mass (m_b)	1.22 kg
Bar radius (r_b)	0.03 m
Cable stiffness (k)	2300 N/m
Rest length of spring (l_0)	0.065 m
Simulation timestep (Δt)	0.001 s
Max motor speed	20 cm/s
Pre-tension	25 N

The simulation environment includes a flat ground surface with friction coefficient $\mu = 0.5$. The robot's initial configuration is set to a three-contact (stance) phase, and the control inputs are generated using the iLQR algorithm to track the reference trajectory derived from the critical configuration search.

The simulation results demonstrate the effectiveness of the proposed control framework in achieving continuous rolling locomotion. Fig. 6 shows the trajectory of the robot's center of mass (GCoM) during the rolling process. The robot successfully transitions between the deformation and rolling phases, as indicated by the alternating three-contact and two-contact states.

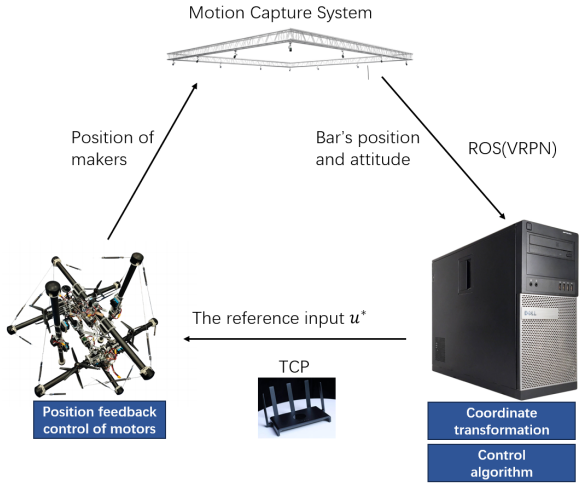

Fig. 7. The physical experimental environment.

B. Physical experiment

Physical experiment setup: The physical experiment system (Fig. 7) comprises a six-bar tensegrity prototype equipped with 24 servo motors, a computer loaded with the closed-loop control algorithm, an '*optitrack*' motion capture system and a router. The prototype's data is transmitted to Robot Operating System (ROS) through Virtual Reality Peripheral Network (VRPN) with a 240 Hz transmission frequency. The inverse dynamics based rolling control algorithm is implemented on MATLAB 2023a software. The control signals generated by the algorithm are sent to the control board of motors via WiFi.

Physical experiment results: Similar to the simulation setup, we predefined a continuous rolling sequence consisting of 12 contact transitions in the physical experiment. Due to hardware limitations, the prototype was constrained to a maximum motor speed of 1 cm/s and a control frequency of 20 Hz. Fig. 8 illustrates the contact sequence and the CoM trajectory, where the overlapping CoM positions result from post-collision oscillations. Despite these disturbances, the experimental results demonstrate that the prototype successfully completed all 12 predefined rolling motions under the controller's guidance. Fig. 9 shows the cable length variations during the motion. For tensegrity robots of the same size, the cable length variation required for motion is a crucial indicator for evaluating controller performance. Smaller cable length variation indicates higher utilization efficiency of tension force, which can effectively reduce the actuation load. Except for a few jumps caused by collisions, the cable length changes remained below 0.2 m (23% of the bar length), significantly lower than the maximum 30.92% reported in [10]. This improvement validates the effectiveness of the controller's optimized input in reducing cable

actuation effort. The process and results of the physical experiment are provided in the multimedia attachment.

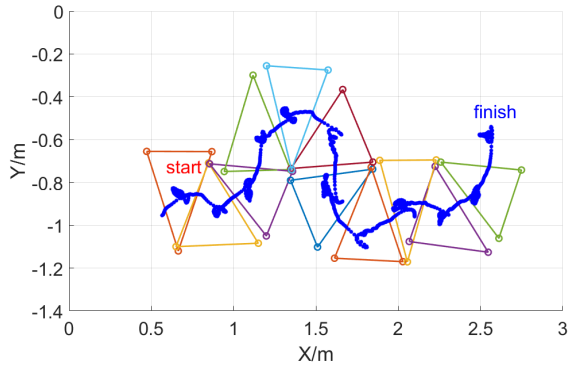

Fig. 8. Trajectory of physical experiments (Top view).

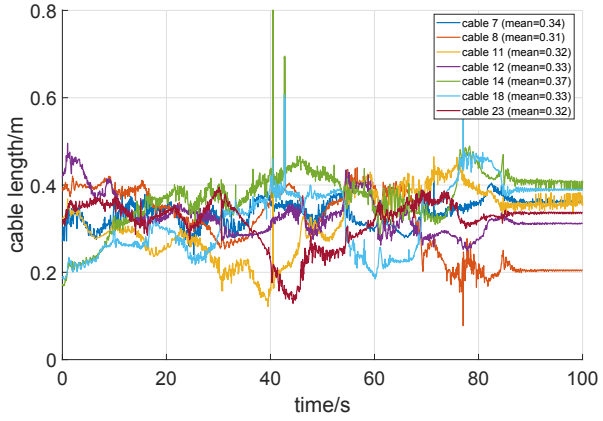

Fig. 9. Control inputs of physical experiments. The legend provides the average length of each cable during the movement process.Among the 24 control inputs of the robot, most play a limited role in rolling. To prevent an excessive number of curves from affecting the readability of the figure, only 7 control inputs with more obvious changes are displayed in the figure.

V. CONCLUSIONS

This paper presents a feasible rolling trajectory generation and control approach for tensegrity robots. Motion decomposition is employed to simplify the state description, avoiding the need for complex full-trajectory tracking. The trajectory generation method is used to obtain the robot's trajectory during the deformation phase, which is then combined with iLQR to achieve continuous rolling control. Simulation and physical experimental results demonstrate that the proposed method can successfully realize continuous rolling control and outperforms existing approaches in terms of input optimization and motion efficiency.

REFERENCES

[1] R. E. Skelton and M. C. de Oliveira, *Tensegrity Systems*. Springer, 2009.

[2] A. P. Sabelhaus et al.,"Tensegrity Robotics: A Review of Structures, Models, and Control," *Soft Robotics*, vol. 7, no. 3, pp. 346 - 361, 2020.

[3] X. Feng, J. Xu, J. Zhang, M. Ohsaki, Y. Zhao, Y. Luo, Y. Chen, and X. Xu, "Trajectory planning on rolling locomotion of spherical movable tensegrity robots with multi-gait patterns," *Soft Robotics*, vol. 11, no. 5, pp. 725–740, 2024.

[4] Q. Yang, S. Hao, Q. Liu, S. Liu, and H. Fang, "Dynamic modeling and control for a collision-resilient tensegrity aerial vehicle," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2025.

[5] S. Liu, Z. Jing, S. Hao, J. Lyu, Z. Tao, Y. Gui, H. Fang, and Q. Yang, "Joint terrestrial-aerial path planning for tensegrone robot," *Unmanned Systems*, 2025.

[6] M. Zhang et al., "Deep Reinforcement Learning for Tensegrity Robot Locomotion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[7] B. T. Mirletz et al., "Goal - Directed CPG - Based Control for Tensegrity Robots," *Soft Robotics*, vol. 2, no. 4, pp. 165 - 176, 2015.

[8] S. Pellegrino, "Mechanics of Kinematically Indeterminate Structures," *University of Cambridge*, 1986.

[9] J. M. Tur and S. H. Juan, "Tensegrity Frameworks: Dynamic Analysis Review and Open Problems," *Mechanism and Machine Theory*, vol. 44, no. 1, pp. 1 - 18, 2009.

[10] R. Motro, "*Tensegrity: Structural Systems for the Future*". Kogan Page Science, 2003.

[11] W. Du et al., "Force Analytic Method for Rolling Gaits of Tensegrity Robots," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2249 - 2259, 2016.

[12] K. Kim et al., "Rolling Locomotion of Cable - Driven Soft Spherical Tensegrity Robots," *Soft Robotics*, vol. 7, no. 3, pp. 346 - 361, 2020.

[13] L. - H. Chen et al., "Inclined Surface Locomotion Strategies for Spherical Tensegrity Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[14] F. Xiaodong, F. Yangbiao, F. Da, Z. Wenyan, and Z. Yiwen, "Novel nodal relaxation algorithm with its application on nonlinear analysis of tensegrity-based structures," in *Structures*, vol. 58. Elsevier, 2023, p. 105272.

[15] J. Jeong, I. Kim, Y. Choi, S. Lim, S. Kim, H. Kang, D. Shah, R. Baines, J. W. Booth, R. Kramer-Bottiglio *et al.*, "Spikebot: A multigait tensegrity robot with linearly extending struts," *Soft Robotics*, vol. 11, no. 2, pp. 207–217, 2024.

[16] A. Iscen et al., "Flop and Roll: Learning Robust Goal - Directed Locomotion for a Tensegrity Robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[17] B. T. Mirletz et al., "Goal - Directed CPG - Based Control for Tensegrity Robots," *Soft Robotics*, vol. 2, no. 4, pp. 165 - 176, 2015.

[18] A. Iscen et al., "Controlling Tensegrity Robots Through Evolution," *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, 2013.

[19] M. Zhang et al., "Deep Reinforcement Learning for Tensegrity Robot Locomotion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[20] J. Luo et al., "Tensegrity Robot Locomotion Under Limited Sensory Inputs via Deep Reinforcement Learning," *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[21] Y. Guo and H. Peng, "Full-actuation rolling locomotion with tensegrity robot via deep reinforcement learning," in *2021 5th International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 2021, pp. 51–55.

[22] N. J. Kong, C. Li, G. Council, and A. M. Johnson, "Hybrid iLQR model predictive control for contact implicit stabilization on legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4712–4727, 2023.

[23] K. Li, M. Tucker, R. Gehlhar, Y. Yue, and A. D. Ames, "Natural multicontact walking for robotic assistive devices via musculoskeletal models and hybrid zero dynamics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4283–4290, 2022.

[24] S. Liu, Q. Yang, J. Lv, and H. Fang, "Modeling of a six-bar tensegrity robot using the port-hamiltonian framework and experimental validation," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4439–4446, 2024.

[25] K. Koohestani, "On the Analytical Form - Finding of Tensegrities," *Composite Structures*, vol. 166, pp. 114 - 119, 2017.

[26] Y. Wang et al., "Form - Finding of Tensegrity Structures via Rank Minimization of Force Density Matrix," *Engineering Structures*, vol. 227, p. 111419, 2021.

[27] A. B. M. Cera, *Design, Control, and Motion Planning of Cable-Driven Flexible Tensegrity Robots*. University of California, Berkeley, 2020.