Physics-Informed Convex MPC for Real-Time Locomotion of a Tendon-Driven Soft Quadruped

Saumya Karan[1], Neerav Maram[2], Suraj Borate[1], Madhu Vadali[1]

***Abstract*—We present a physics-informed convex optimization framework for real-time whole-body locomotion control of SLOT (Soft Legged Omnidirectional Tetrapod), a tendon-driven soft quadruped with four 3D-printed TPU legs. A central difficulty in controlling compliant legged robots is that the governing leg dynamics are nonlinear, infinite-dimensional PDEs, which are incompatible with real-time convex solvers. We resolve this via a three-level optimization decomposition: (i) a space-discretized Cosserat rod model that converts continuum leg PDEs into a tractable finite-dimensional ODE, capturing large deformations, distributed elasticity, tendon actuation, and penalty-based ground contact; (ii) a modular decoupled formulation aggregates per-leg reaction forces to restore the linear torso prediction structure required for convex MPC; and (iii) a physics-derived closed-form force-to-tendon mapping makes the actuation inversion an analytic, convex operation. The resulting convex quadratic program (QP) is solved at 30 Hz on an NVIDIA Jetson Xavier, achieving less than 5 mm RMSE in center-of-mass trajectories across crawling, walking, and omnidirectional gaits, and empirically stable behavior under six classes of structured perturbations with zero infeasible QP steps across all deterministic perturbation scenarios. Video is available at: Youtube Video**

Keywords — Soft Quadruped, Tendon-driven, Cosserat-Based Modeling, Dynamics, Model Predictive Control, Convex Optimization

I. Introduction

Convex model predictive control (MPC) has become a standard framework for legged locomotion [1], [2]: the linear time-varying structure of rigid-body dynamics allows trajectory optimization to be cast as a real-time-solvable quadratic program (QP). Extending this success to soft-bodied robots is non-trivial. Compliant legs introduce nonlinear, distributed deformation governed by coupled PDEs — breaking the linear prediction structure and making the plant incompatible with direct convex formulations. Existing soft quadruped platforms either sidestep this with purely learned controllers [3], [4], simplified constant-curvature kinematics [5], or sacrifice real-time tractability [6], [7].

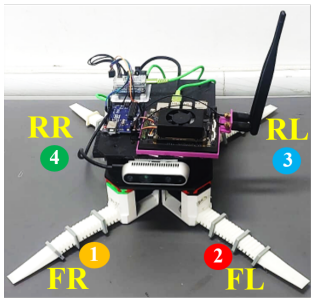
The present work addresses three open questions for the optimization community: (1) Can a physics-based soft-leg model be made fast enough for embedded real-time simulation without sacrificing physical fidelity? (2) Can that model be interfaced with a convex MPC without re-introducing nonlinear coupling at the controller level? (3) Does the resulting convex QP remain feasible in practice across diverse gaits and perturbations?

We answer all three questions through a physics-informed optimization framework: (i) a space-discretized Cosserat rod formulation converts the nonlinear continuum leg PDE into a finite-dimensional ODE, enabling ∼0.35 ms per-leg simulation on embedded hardware; (ii) a modular decoupled coupling isolates per-leg nonlinear dynamics from global trajectory optimization, restoring the linear structure required for convex MPC; (iii) and experiments demonstrate empirical recursive feasibility and stable closed-loop behavior across six perturbation scenarios on a physical soft quadruped.

This work shows that soft-body dynamics, typically considered incompatible with convex optimization, can be reformulated through a force-level interface that preserves convexity without simplifying the underlying physics.

II. System Overview

SLOT (Soft Legged Omnidirectional Tetrapod), shown in Fig. 1, comprises a rigid cuboid torso ($125.5 \times 85.5 \times 34$ mm, $m_c = 2.0$ kg) supported by four 3D-printed TPU legs (40 g each), for a total mass of 2.16 kg [13]. Each leg is actuated by a single Dynamixel XL430 servo (1.4 Nm) via a tendon routed through a virtual pulley at 168° from horizontal. All computation is onboard (NVIDIA Jetson Xavier, 8 GB RAM, 384-core Volta GPU); an Intel RealSense D435 provides visual-inertial odometry via a ROS2 state-estimation node. No off-board computation is used.



FR: Front Right
FL: Front Left
RL: Rear Left
RR: Rear Right

Fig. 1: SLOT robot hardware

III. Cosserat Rod Leg Model

A. From PDE to Tractable ODE

Cosserat rod theory [8], [9] models slender elastic bodies through momentum balance along arc-length $s \in [0, L]$, yielding coupled nonlinear PDEs that are impractical to embed in a real-time controller. We apply a space discretization

[1]Indian Institute of Tecnology, Gandhinagar, India {karansaumya, surajb, madhu.vadali}@iitgn.ac.in
[2]Sreenidhi Institute of Science and Technology, India 22311a1901@ecm.sreenidhi.edu.in

Code and CAD: SLOT_Github. Supported by the Prime Minister's Research Fellowship (PMRF), Government of India (No. OTH/12205).

following [10], [11]: the arc-length domain is divided into $N = 31$ nodes with uniform spacing $\Delta s = L/(N-1)$ ($L = 0.19\,\text{m}$), converting the infinite-dimensional PDE into a finite-dimensional ODE system. Each leg is modeled in the planar XZ frame — dominant bending direction for all tested gaits. Each node i carries generalized coordinates

$$\mathbf{q}_i = \begin{bmatrix} x_i & z_i & \theta_i \end{bmatrix}^\top, \tag{1}$$

corresponding to planar position and cross-section orientation.

B. Internal Elastic Forces and Bending Moments

For each segment between nodes i and $i{+}1$, the axial strain is

$$\varepsilon_i = \frac{l_i - \Delta s}{\Delta s}, \quad l_i = \sqrt{(x_{i+1}-x_i)^2 + (z_{i+1}-z_i)^2}. \tag{2}$$

The internal axial force is $f_{\text{ax},i} = E_i A_i \varepsilon_i$, where $E_i = 10^7\,\text{Pa}$ (TPU Young's modulus), $A_i = w \cdot h_i$ with width $w = 20\,\text{mm}$ and linearly tapered thickness $h_i \in [3.5, 13.5]\,\text{mm}$. Forces are projected into the global frame via the local rotation matrix $\mathbf{R}_i$ and applied in opposite directions at nodes i and $i{+}1$:

$$\begin{bmatrix} F_{x,i} \\ F_{z,i} \end{bmatrix} \leftarrow \begin{bmatrix} F_{x,i} \\ F_{z,i} \end{bmatrix} + \mathbf{R}_i^\top \begin{bmatrix} +f_{\text{ax},i} \\ 0 \end{bmatrix}, \quad \mathbf{R}_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}. \tag{3}$$

Bending is captured by the relative orientation change $\Delta\theta_i = \theta_{i+1} - \theta_i$, giving internal moments

$$m_{\text{bend},i} = \frac{E_i I_i}{\Delta s}\,\Delta\theta_i, \quad I_i = \frac{w\,h_i^3}{12}, \tag{4}$$

applied as equal and opposite torques at nodes i and $i{+}1$.

C. External Loads

Ground contact is modeled as a penalty spring-damper at each node:

$$F_{\text{cnt},i} = \begin{cases} k_c\,\delta_i + d_c\,\dot{\delta}_i & z_i < 0, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

with $k_c = 10^5\,\text{N/m}$, $d_c = 10\,\text{Ns/m}$, and penetration $\delta_i = -z_i$. Coulomb friction resists sliding at contact nodes: $F_{\text{fric}} = -\mu(m_{\text{leg}} + M_c/4)\,g\,\text{sign}(v_x)$, $\mu = 0.1$. A time-varying tendon force $T(t)$ with a ramp–hold–decay profile (peak $T_{\max} = 1.5\,\text{Nm}$, max angle 116.28°) is applied along the proximal $\frac{2}{3}$ of each leg. Linear damping ($c = 0.1$) and a virtual restoring spring ($k_r = 800\,\text{N/m}$) provide numerical stabilization.

D. Assembled Node-Level Dynamics

Superimposing all loads yields the standard robot-dynamics form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{T}(\mathbf{q},t), \tag{6}$$

with $\mathbf{M} = \text{diag}(m_0, m_0, I_0, \ldots, m_N, m_N, I_N)$ (lumped node masses and inertias), $\mathbf{C}\dot{\mathbf{q}}$ encoding damping and restoring forces, and $\mathbf{T} = \mathbf{F}_{\text{int}} + \mathbf{F}_{\text{cnt}} + \mathbf{F}_{\text{fric}} + \mathbf{F}_{\text{tendon}}$. Explicit Euler integration of (6) at $\Delta t = 10^{-4}\,\text{s}$ requires ≈0.35 ms per leg on the Jetson Xavier — over 100× faster than a comparable FEM formulation [7]. The single leg deformation under tendon actuation is in our work [15].

IV. Whole-Body Modular Coupling

The torso is modeled as a rigid cuboid with six DOF. Each leg is simulated independently as a planar Cosserat rod; the tip reaction force $\mathbf{F}_i$ at attachment point $\mathbf{r}_i$ is passed to the torso as shown in Fig. 2. Neglecting gyroscopic precession terms (valid for low angular velocities in walking and crawling, as in [12]), the torso dynamics are:

$$m_c\ddot{\mathbf{p}} = \sum_{i=1}^{4} \mathbf{F}_i, \qquad \mathbf{I}_c\dot{\boldsymbol{\omega}} = \sum_{i=1}^{4} (\mathbf{r}_i \times \mathbf{F}_i). \tag{7}$$

This modular structure is the key optimization enabler: the nonlinear leg simulation is confined to a separate integration step, while (7) is linear in the forces $\mathbf{F}_i$ and can be linearized into a convex prediction model for the MPC.

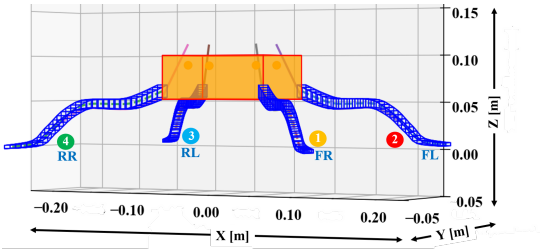


Fig. 2: Force application from four Cosserat rod legs onto the rigid cuboid torso.

V. Convex MPC Formulation

The key observation is that the torso dynamics are affine in the contact forces, while the nonlinear Cosserat rod dynamics are evaluated outside the optimization loop. Thus, the MPC prediction model remains linear in decision variables, and the resulting optimization problem is a convex quadratic program. The MPC algorithm integrates a predictive model derived from Cosserat rod dynamics and rigid body torso dynamics to maintain desired trajectories and velocities. The control architecture consists of a hierarchical feedback loop for legged locomotion. An overview of the control architecture is shown in Fig. 3.

For clarity and reproducibility, Algorithm 1 in Section IX (Appendix) outlines the full closed-loop execution sequence implemented on the physical robot.

A. State, Input, and Prediction Model

The robot state is $\mathbf{x}_k = [\phi, \theta, \psi, p_x, p_y, p_z, \omega_x, \omega_y, \omega_z, v_x, v_y, v_z]^\top \in \mathbb{R}^{12}$ and the control input $\mathbf{u}_k \in \mathbb{R}^{12}$ collects three-axis ground reaction forces for the four legs. Linearizing (7) about $\mathbf{x}_k$ and discretizing at 30 Hz yields a linear time-varying model $\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k$, structurally identical to rigid-body MPC [1]. The decoupled Cosserat simulations supply the forces $\mathbf{F}_i$ that define $\mathbf{B}_k$, bridging soft-body mechanics and the convex controller without any approximation to the leg model itself.

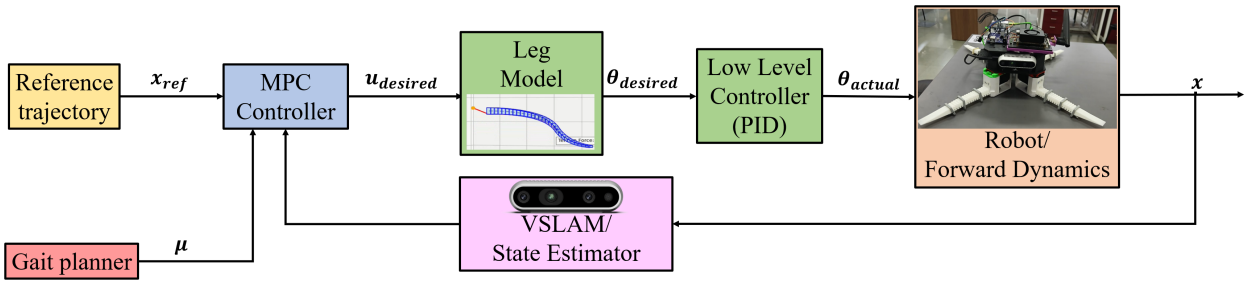


Fig. 3: Model Predictive Control block diagram.

B. Convex QP

The MPC solves at each 30 Hz control step:

$$\min_{\{\mathbf{u}_k\}_{k=0}^{H-1}} \sum_{k=0}^{H-1} \Big[(\mathbf{x}_k - \mathbf{x}_k^{\text{ref}})^\top \mathbf{Q} (\mathbf{x}_k - \mathbf{x}_k^{\text{ref}}) + \mathbf{u}_k^\top \mathbf{R}\, \mathbf{u}_k \Big] \quad (8)$$

with horizon $H = 15$ (≈ 0.5 s), subject to

$$f_{z,i} \le 6.0\ \text{N}, \quad \textstyle\sum_{i=1}^{4} f_{z,i} = m_c g, \quad (9)$$

$$|f_{x,i}|,\ |f_{y,i}| \le \mu_i f_{z,i}, \quad \forall\, i. \quad (10)$$

Constraint (9) enforces static force balance; (10) implements gait-phase-varying linearized friction cones ($\mu_i \in \{0.2, 0.6\}$). The weighting matrices

$$\mathbf{Q} = \tfrac{1}{1000}\,\text{diag}(0.1, 0.1, 0.1,\ 0, 0, 100,\ 0.01, 0.01, 0.01,\ 20, 0.1, 0),$$
$$\mathbf{R} = 10^{-8}\mathbf{I}_{12},$$

prioritize vertical body support (p_z, weight 100) and forward velocity (v_x, weight 20). The QP is solved via SCS through CVXPY in ≈ 6.2 ms per update, well within the 33 ms control budget. The gait phase scheduler modulates μ_i and contact constraints, encoding stance/swing assignment as time-based switching.

C. Physics-Informed Force-to-Tendon Mapping

The MPC outputs desired normal forces $f_{z,i}^*$. Converting these to servo angles is a nonlinear actuation inversion problem. We exploit the Cosserat rod simulation offline: sweeping tendon angles under quasi-static single-leg loading yields the empirical affine fit

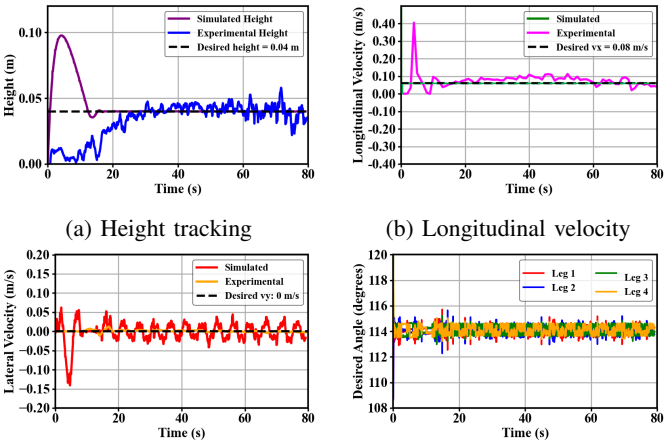
$$\theta_i\ [\text{deg}] = \frac{f_{z,i} + 6.91}{0.107}, \quad (11)$$

validated independently of full-body locomotion data. This mapping is linear in $f_{z,i}$, preserving the convex structure of the pipeline end-to-end. Residual dynamics (tendon compliance, TPU hysteresis, motor lag) are handled implicitly by the receding-horizon MPC updates and onboard PID tracking.

VI. Experimental Results

A. Validation Setup

ArUco markers placed at the torso CoM and four points along each leg (fixed base, 1/3, 2/3, tip) were tracked by a RealSense D435 (depth uncertainty <2 mm at 0.6–1.0 m). Simulation was driven by the same tendon schedules used on hardware, with no parameter re-tuning on full-body data.



(a) Height tracking (b) Longitudinal velocity

(c) Lateral velocity (d) Desired joint angles

Fig. 4: Experimental and simulated results of quadruped MPC for the walking gait. Leg 1: FL, Leg 2: RL, Leg 3: RR, Leg 4: FR.

B. Leg and Whole-Body Accuracy

[15] reports single-leg vertical-position errors across four representative nodes; accuracy exceeds 95% at all nodes. The higher NRMSE at the tip (19.2%) reflects normalization over a small absolute displacement range, not large absolute errors. For whole-body motion, the CoM RMSE is 4.0 mm (x) and 4.8 mm (z) for forward walking, and 5.1 mm (x) and 4.6 mm (z) for crawling, demonstrating consistent tracking performance across gaits.

C. MPC Feasibility and Stability

The MPC tracks height $p_z^* = 0.04$ m and forward velocity $v_x^* = 0.08$ m/s during walking; lateral velocity remains near zero throughout as shown in Fig. 4.

Additionally, Fig. 5 provides a time-lapse sequence of the robot walking in a straight line under MPC control, demonstrating stable forward locomotion and visually confirming the controller's effectiveness in maintaining gait consistency. Hardware experiments over o.8 m confirmed <3.2 cm maximum lateral drift, primarily from unmodeled friction asymmetry.

For a 60° omnidirectional gait ($v_x^* = 0.052$ m/s, $v_y^* = 0.09$ m/s, matching $\tan 60°$), path, as shown in Fig. 6 RMSE is 3.5 cm.

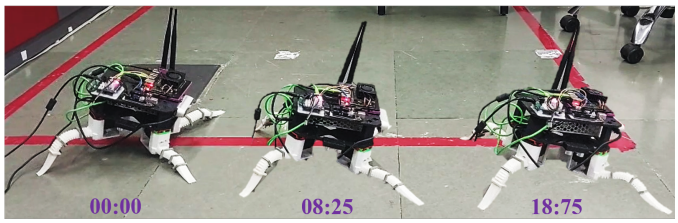
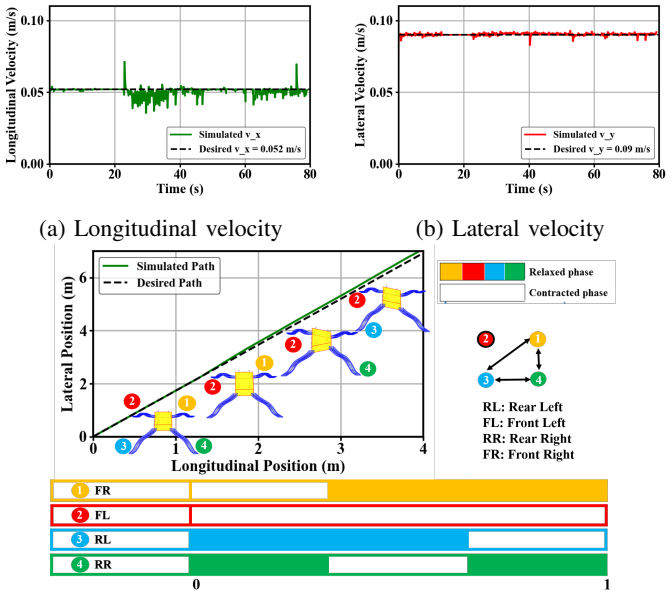


Fig. 5: SLOT hardware robot walking in a straight line under MPC control, demonstrating stable forward locomotion.



(a) Longitudinal velocity (b) Lateral velocity

(c) Desired versus simulated path with desired gait sequence. Colored circles near the legs indicate leg contraction.

Fig. 6: Simulation results for the 60° omnidirectional gait under MPC.

Table I reports stability metrics over a 25 s horizon for seven perturbation scenarios (walk gait). All six deterministic cases achieve zero final MPC cost and settle within 14.6 s with zero infeasible QP steps. The persistent Gaussian noise case ($\sigma = 0.001$ on all states) does not converge — expected for a non-vanishing disturbance — but remains bounded, consistent with practical input-to-state stability. Figure 7 illustrates the combined MPC cost trajectories. A Dynamic Constraint Map (DCM), depicted in Figure 8, confirms uniform feasibility for the walk gait; the omnidirectional gait has two brief system-level violations at $t = 22.4$ s and $t = 61.9$ s during phase transitions, resolved by reverting to the last feasible input.

VII. Discussion and Future Directions

The key structural insight is that the source of nonlinearity (soft-leg deformation) and the target of optimization (torso trajectory) are separable at the force interface. By running the nonlinear Cosserat simulation per-leg and passing only the resulting forces to the linear torso model, the global prediction model is linear and the MPC reduces to a convex QP. The physics-derived

TABLE I: MPC stability metrics — walk gait, 25 s horizon (all deterministic cases: zero infeasible QP steps).

Scenario	Max Cost	Mean Cost	Final Cost	Settle (s)
Baseline	4.97	1.17	0.00	14.55
Roll	4.97	1.17	0.00	14.55
Pitch	4.79	1.11	0.00	13.70
Height	6.22	1.54	0.00	14.55
Velocity	2.38	0.17	0.00	10.10
Combined	3.70	0.79	0.00	12.61
Noise	2.38	0.22	0.49	—

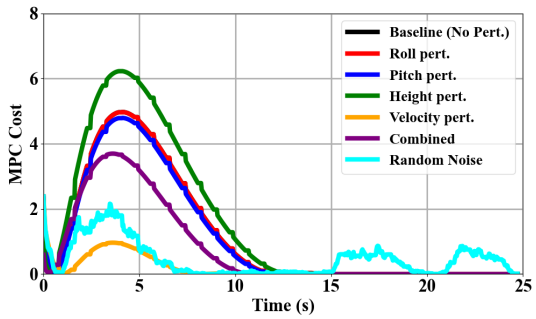
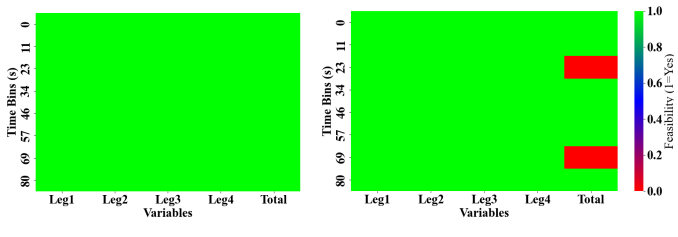


Fig. 7: Combined MPC cost trajectories under perturbations for Walk gait. Baseline and Roll overlap; Noise exhibits non-settling oscillations.

affine map (11) then closes the actuation loop without breaking this structure. This decomposition principle may generalize to compliant multi-body systems in which deformation-induced forces can be precomputed or co-simulated, enabling similar convex reformulations. The proposed decomposition is not specific to tendon-driven quadrupeds and may extend to other compliant robotic systems where internal deformation can be decoupled from global rigid-body motion.

Future directions Embedding the discretized Cosserat model in an auto-differentiable framework would enable gradient-based trajectory optimization and parameter identification. Learned policies can provide near-optimal warm-starts to reduce MPC solve time and enable longer horizons. The current formulation assumes quasi-static force-to-tendon mapping and does not explicitly model actuator dynamics or contact uncertainty. Additionally, theoretical guarantees on recursive



(a) Walk gait (b) Omnidirectional gait

Fig. 8: Dynamic Constraint Map (DCM) illustrating the feasibility status of the MPC solution.

feasibility remain an important direction for future work.

VIII. Conclusion

We presented a physics-informed convex optimization framework for tendon-driven soft quadruped locomotion. By combining discretized Cosserat rod simulation with modular force-level coupling, the approach preserves the linear structure required for convex MPC without simplifying soft-body dynamics. Experiments at 30 Hz demonstrate <5 mm CoM tracking accuracy, empirical recursive feasibility, and zero infeasible QP steps across structured perturbations. This work highlights physics-informed decomposition as a pathway for extending convex optimization to compliant robotic systems.

References

[1] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, 2018.

[2] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2017.

[3] S. Bhagat, H. Banerjee, Z. T. H. Tse, and H. Ren, "Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges," *Robotics*, vol. 8, no. 1, p. 4, 2019.

[4] Y. Li, Y. Li, T. Ren, J. Xia, H. Liu, C. Wu, S. Lin, and Y. Chen, "An untethered soft robotic dog standing and fast trotting with jointless and resilient soft legs," *Biomimetics*, vol. 8, no. 8, p. 596, 2023.

[5] X. Huang, J. Zou, and G. Gu, "Kinematic modeling and control of variable curvature soft continuum robots," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 6, pp. 3175–3185, 2021.

[6] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Syst. Mag.*, vol. 43, no. 3, pp. 30–65, 2023.

[7] T. M. Bieze, C. Della Santina, L. A. Roché, M. Cianchetti, and D. Rus, "Finite element method-based kinematics and closed-loop control of soft continuum manipulators," *Soft Robot.*, vol. 5, no. 3, pp. 348–364, 2018.

[8] L. Xun, G. Zheng, and A. Kruszewski, "Cosserat-rod-based dynamic modeling of soft slender robots interacting with the environment," *IEEE Trans. Robot.*, vol. 40, pp. 2811–2830, 2024.

[9] A. A. Alqumsan, S. Khoo, and M. Norton, "Robust control of continuum robots using Cosserat rod theory," *Mech. Mach. Theory*, vol. 131, pp. 48–61, 2019.

[10] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[11] M. Gazzola, L. D. van Rees, and P. Koumoutsakos, "Forward and inverse problems in the mechanics of soft filaments," *Roy. Soc. Open Sci.*, vol. 5, no. 3, 2018.

[12] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Auton. Robots*, vol. 41, no. 1, pp. 259–272, 2017.

[13] S. Karan, R. Harikrishnan, Y. Dodia, T. Phopalkar, S. Borate, and M. Vadali, "SLOT: A soft-legged omnidirectional tetrapod," in *Walking Robots into Real World (CLAWAR 2024)*, Lect. Notes Netw. Syst., vol. 1115. Cham: Springer, 2024.

[14] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1728–1748, 2023.

[15] S. Karan, N. Maram, S. Borate, and M. Vadali, "Dynamic Modeling and MPC for Locomotion of Tendon-Driven Soft Quadruped," *arXiv preprint arXiv:2602.16371*, 2026.

IX. Appendix

For clarity and reproducibility, Algorithm 1 outlines the full closed-loop execution sequence implemented on the physical robot.

Algorithm 1 ROS2-Based Hardware-in-the-Loop MPC Locomotion Pipeline for SLOT

System Initialization:
Initialize ROS2 middleware and time synchronization
Load robot geometric, inertial, and Cosserat rod leg parameters
Initialize onboard computation for perception and control
Configure publishers and subscribers for all control nodes
Perception and Localization:
Start visual–inertial SLAM using RGB-D and IMU inputs
Publish robot odometry and pose estimates on `/odom`
Broadcast frame transformations via `/tf`
State Estimation Node:
Subscribe to `/odom` and inertial measurements
Extract position, orientation, linear and angular velocities
Construct MPC state vector

$$\boldsymbol{x}_k = [\phi, \theta, \psi,\ p_x, p_y, p_z,\ \omega_x, \omega_y, \omega_z,\ v_x, v_y, v_z]$$

Publish estimated state on `/mpc_state`
Controller Initialization:
Initialize gait scheduler and phase manager
Initialize MPC controller with horizon H, timestep Δt, and cost weights
Initialize low-level tendon motor controllers
while robot is operational **do**
  State Update:
  Receive latest robot state $\boldsymbol{x}_k$ from `/mpc_state`
  Gait and Reference Generation:
  Update gait phase (walk, crawl, omnidirectional)
  Determine stance and swing legs
  Generate desired body height and velocity references
  Form reference trajectory $\boldsymbol{x}_{\mathrm{ref},k}$
  MPC Optimization:
  Linearize decoupled torso dynamics about $\boldsymbol{x}_k$
  Formulate convex MPC using predicted ground reaction forces
  Enforce force balance, friction cone, and contact constraints
  Solve for optimal control input

$$\boldsymbol{u}_k^\star = [f_{x,1}, f_{y,1}, f_{z,1}, \ldots, f_{x,4}, f_{y,4}, f_{z,4}]$$

  if optimization is infeasible **then**
    Apply last feasible control input
  end if
  Force-to-Actuation Mapping:
  for $i = 1$ to 4 **do**
    Convert desired normal force $f_{z,i}$ to tendon angle θ_i
    Use precomputed Cosserat rod–based force–angle relation
  end for
  Publish desired tendon angles on `/desired_joint_angles`
  Low-Level Motor Control:
  Subscribe to `/desired_joint_angles`
  Track tendon angles using PID control
  Compensate actuator and tendon dynamics via feedback
  Execution and Feedback:
  Apply motor commands to the robot
  Execute compliant leg motion
  Advance control loop by Δt
end while