EquiMus: Energy-Equivalent Dynamic Modeling and Simulation of Musculoskeletal Robots Driven by Linear Elastic Actuators

Yinglei Zhu, Xuguang Dong, Qiyao Wang, Qi Shao, Fugui Xie, Xinjun Liu, and Huichan Zhao

***Abstract*—Dynamic modeling and control are critical for unleashing soft robots' potential, yet remain challenging due to their complex constitutive behaviors and real-world operating conditions. Bio-inspired musculoskeletal robots, which integrate rigid skeletons with soft actuators, combine high load-bearing capacity with inherent flexibility. Although actuation dynamics have been studied through experimental methods and surrogate models, accurate and effective modeling and simulation remain a significant challenge, especially for large-scale hybrid rigid–soft robots with continuously distributed mass, kinematic loops, and diverse motion modes. To address these challenges, we propose *EquiMus*, an energy-equivalent dynamic modeling framework and MuJoCo-based simulation for musculoskeletal rigid–soft hybrid robots with linear elastic actuators. The equivalence and effectiveness of the proposed approach are validated and examined through both simulations and real-world experiments on a bionic robotic leg. EquiMus further demonstrates its utility for downstream tasks, including controller design and learning-based control strategies.**

***Index Terms*—Modeling, control, and learning for soft robots, biologically-inspired robots, dynamics, simulation and animation.**

I. Introduction

BIO-INSPIRED robots have been extensively studied and developed in recent years [1], [2], [3], [4], [5]. Among them, articulated musculoskeletal robots feature rigid skeletons and soft artificial muscles, enhancing the robot's load capacity and intrinsic adaptability [6], [7], [8]. In these systems, elastic actuators (EAs) boast high energy density and inherent compliance [9]. Dynamic modeling and simulation of these systems lay important foundations for their design, control [10], and data generation [11].

However, classical rigid–body models do not capture configuration-dependent mass redistribution, large elastic strains with damping, or loop-closure constraints in rigid–soft hybrids [12]. In practice, soft actuators often have non-negligible mass in load-bearing limbs, so their inertia must be modeled alongside actuation. As the configuration changes, composite inertia varies, and kinematic loops frequently arise in such systems, both of which complicate the dynamics and make per-robot Lagrangian derivations labor-intensive and hard to generalize.

Meanwhile, rigid-body robotics has advanced rapidly, fueled by physics engines and graphics computation [13]. Simulators such as PyBullet [14], MuJoCo (Multi-Joint dynamics with Contact) [15] and Isaac Gym [16] offer standardized environments well suited for data-driven control and design [17], [18], [19]. Although MuJoCo can mimic viscoelastic behavior [20], [21], it omits actuator inertia, limiting accurate simulation of coupled rigid–soft dynamics. To bridge these gaps, we present *EquiMus*[1], an energy-equivalent modeling and MuJoCo-based simulation framework for rigid–soft hybrid robots with linear elastic actuators. As shown in Fig. 1, EquiMus tackles three challenges: (i) the actuator inertia and variable mass distribution during motion; (ii) loop-closure within musculoskeletal chains; (iii) plug-and-play integration with model-based and learning-based controllers. The main contributions of this work are summarized as:

1) **Modeling**: a compact energy-equivalent lumped-mass formulation that maps elastic actuator dynamics to discrete rigid-body elements, preserving energies and matching the virtual work of damping and actuation;

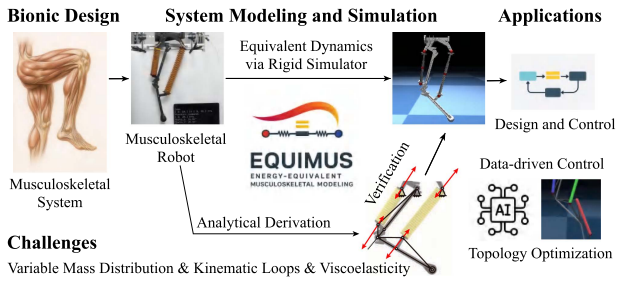


Fig. 1. Overview and motivation of the proposed energy-equivalent modeling framework EquiMus.

Received 30 April 2025; accepted 24 September 2025. Date of publication 15 October 2025; date of current version 30 October 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 52222502 and in part by the Beijing Municipal Natural Science Foundation under Grant E2024202287. This letter was recommended for publication by Associate Editor C. D. Santina and Editor C. Laschi upon evaluation of the reviewers' comments. *(Corresponding author: Huichan Zhao.)*

The authors are with the Department of Mechanical Engineering, State Key Laboratory of Tribology in Advanced Equipment, Beijing Key Laboratory of Transformative High-end Manufacturing Equipment and Technology, Tsinghua University, Beijing 100084, China (e-mail: zhaohuichan@mail.tsinghua.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3621980, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3621980

[1]All code, data and derivation used in this letter are available at https://github.com/fly-pigTH/EquiMus

2) **Implementation**: a MuJoCo realization with loop-closure constraints, smooth elastic actuator dynamics, and out-of-the-box reinforcement learning (RL) compatibility;
3) **Validation and Applications**: experiments on a pneumatic leg with fluidic elastomer actuators (FEAs), show close sim-to-real agreement and enable PID auto-tuning, model-based control, and RL-based control.

II. Literature Review

A. Dynamic Modeling of Soft Robots

Modeling soft robots remains challenging due to inherent continuum properties, nonlinear material behaviors, and high-dimensional configuration spaces. Among them, articulated soft robots, which are rigid–soft hybrids with discrete links and soft actuation, represent a distinct subclass that requires tailored modeling considerations. Existing methods can be categorized into four mainstream approaches [22]:

Continuum mechanics models (e.g., Cosserat rod and nonlinear Euler-Bernoulli beam formulations [23]) deliver high fidelity for slender, continuously deformable structures but incur prohibitive computation, making them unsuitable for real-time control in articulated hybrids.

Geometrical models (e.g., piecewise-constant-curvature [24]) approximate deformations via low-dimensional curves, trading expressiveness for computational speed for continuum manipulators, but limited for multi-link hybrids.

Discrete material models condense mass and compliance into discrete elements [25], balancing accuracy and efficiency—suited for articulated hybrids with distributed elasticity.

Surrogate models apply neural networks (NNs) to model nonlinear behaviors [26] but demand extensive training data, relying on fast simulators for dataset generation.

Accordingly, we adopt a discrete (lumped-mass) approach, because it preserves actuator inertia at real-time rates and strikes the desired fidelity-efficiency balance.

B. Simulation of Soft Robots

Hand-crafting dynamic models for soft/rigid–soft systems is costly, and coupling them with modern controllers (e.g., reinforcement learning) is labor-intensive. With advances in biomechanics and graphics, simulators have become the default for design, analysis, and data generation.

Early PyBullet-based, continuum-oriented efforts such as SoMo [12] and SoMoGym [27] approximate soft manipulators via segment discretization with beam theory. These methods capture quasi-static behavior effectively but lack full dynamics. Moreover, because the Unified Robot Description Format (URDF) assumes open chains, they cannot represent the loop closures [28] that are common in articulated musculoskeletal robots. More physics-rich simulators (e.g., SoftManiSim [29] and Jitosho's framework [30]) incorporate advanced Cosserat/beam theory and demonstrate RL-based locomotion. However, they are typically confined to centimeter-scale tasks and incur significant computational cost, limiting real-time control studies. FEM-based SOFA (Simulation Open Framework Architecture) [31] offers high physical fidelity for soft tissues and continuum structures but is likewise heavy for real-time applications.

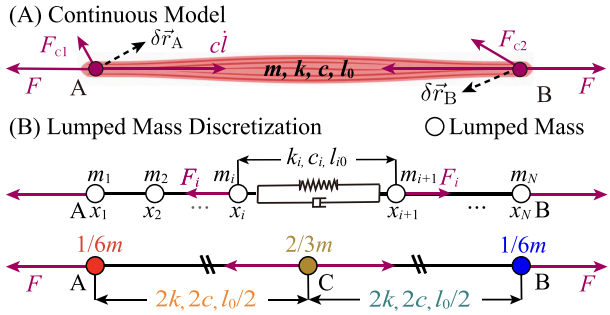


Fig. 2. Schematic diagram of the soft actuator and its lumped mass distribution. The actuator is subjected to gravity, driving forces F, elastic forces, viscous resistance $c\dot{l}$ at endpoints, and constraint forces F_c from the rigid skeletons. $\delta\vec{r}$ denotes the virtual displacement. Since F_c is internal to the system, it is excluded from the Lagrangian dynamics. Driving forces and viscous forces are defined as positive as shown in figure.

By contrast, MuJoCo [15] has become a preferred platform for articulated systems and has been largely used for model-based/data-driven control and sim-to-sim testing of rigid-body robots [32], [33]. Its XML format supports kinematic loops and musculoskeletal simulations [20], [21]. However, MuJoCo's actuators are still modeled as massless elements, limiting its ability to capture dynamic mass redistribution (DMR)-i.e., changes in the mass distribution of embedded soft actuators during motions. OpenSim [34], [35], [36] shares this limitation.

Classic simulators are summarized in Table I. It highlights key capabilities such as support for kinematic loops and the ability to handle dynamic mass redistribution. In summary, off-the-shelf simulators still have limitations in realizing the dynamic modeling and simulations for musculoskeletal robots.

III. Methods

A. Basic Assumptions

An elastic actuator, illustrated in Fig. 2, has endpoints A, B, mass m, stiffness k, damping c, rest length l_0, current length l, and driving force **F**. The actuator is assumed to act as a driving unit in the rigid–soft hybrid robot and follow basic conditions:

- **Unidirectional and Uniform Deformation**: The actuator, with a negligible radial size, is modeled as a one-dimensional linear element with a uniform mass distribution, with a linear density, denoted as $\rho = m/l$.
- **Axial Driving Forces**: The actuator is subjected to a pair of equal and axial forces, denoted as **F**, applied at endpoints A and B along its axis.
- **Spring-Mass-Damper System**: With one end fixed, we assume that the actuator behaves as a second-order system with constant elasticity and viscosity [37] [38].

TABLE I
COMPARISON OF DIFFERENT SIMULATION FRAMEWORKS

Framework	Modeling Approach	Physics Complexity	Kinematic Loop	R–S Hybrid Robots	DMR
PyBullet/Gazebo	rigid-body	low	×	×	×
Webots	rigid-body	low	×	×	×
Elastica	Cosserat rods	medium	×	×	✓
SOFA	FEM, finite element method	high	✓	✓	✓
SoMo/SoMoGym	rigid-body	low	×	✓	✓
SoftManiSim	rigid-body + Cosserat	medium	×	✓	✓
OpenSim	rigid-body + muscle model	medium-high	✓	✓	×
MuJoCo/MyoSim	rigid-body + muscle model	medium	✓	✓	×
EquiMus*	**rigid-body equivalence**	**low**	✔	✔	✔

B. Energy-Based Perspective on Dynamic Modeling

The robot dynamics are formulated using the vector form of the Lagrangian equation,

$$\left(\frac{d}{dt}\frac{\partial}{\partial \dot{\mathbf{q}}} - \frac{\partial}{\partial \mathbf{q}}\right)(L_{\text{EA}} + L_{\text{other}}) = \mathbf{Q}_{\text{EA}} + \mathbf{Q}_{\text{other}} \tag{1}$$

where L, $\mathbf{q}$, and $\mathbf{Q}$ denote the Lagrangian, generalized coordinates, and generalized forces respectively. We decompose L and $\mathbf{Q}$ into contributions from elastic actuators (EA) and rigid structures (other). L_{other} and $\mathbf{Q}_{\text{other}}$ depend on $\mathbf{q}$, its derivative $\dot{\mathbf{q}}$, and external inputs. From an energy perspective, if the energy and forces of the elastic actuator can be discretized with rigid-body equivalents, the overall dynamics remain invariant, regardless of the specific configuration and type of soft actuators.

C. Energy-Equivalent Model

Inspired by the lumped mass method [22], [25], we discretize the actuator into an assumed energy-equivalent mass-spring-damper array. This formulation integrates the dynamics of the soft actuator into a multi-rigid-body representation, accounting for the elastoplastic behavior, variable mass distribution, and kinematic loops. The theoretical proof is provided through constructive modeling and parameter derivation.

Let $\vec{r}_{\text{A}}$ and $\vec{r}_{\text{B}}$ denote the position vectors of endpoints A and B in the global (inertial) frame, respectively. Suppose N mass points are distributed along the elastic actuator at positions $\vec{x}_1, \ldots, \vec{x}_N$, with corresponding masses $m_1, \ldots, m_N$. Each adjacent pair of points is connected by a linear actuator characterized by stiffness k_i, damping c_i, original length l_{i0}, and internal driving force F_i, where $i = 1, \ldots, N-1$. For convenience, $\vec{x}_i$ is reduced to 1D coordinate ξ_i along the actuator's length, such that. $\vec{x}_i(\xi_i) = (1-\xi_i)\vec{r}_{\text{A}} + \xi_i \vec{r}_{\text{B}}$. In this section, energy quantities in the equivalent model are denoted by $\hat{\cdot}$.

1) Equivalence of Gravitational Potential Energy: The gravitational potential energy of the actuator and its equivalent rigid-body model is given by

$$V_{\text{g}} = \frac{1}{2} m g \vec{k} \cdot (\vec{r}_{\text{A}} + \vec{r}_{\text{B}}) \tag{2}$$

$$\hat{V}_{\text{g}} = \sum\nolimits_{i=1}^{N} m_i g \vec{k} \cdot \vec{x}_i = \sum\nolimits_{i=1}^{N} m_i g \vec{k} \cdot [(1-\xi_i)\vec{r}_{\text{A}} + \xi_i \vec{r}_{\text{B}}] \tag{3}$$

where $\vec{k}$ denotes the unit vector of the ground frame, oriented opposite to gravity.

2) Equivalence of Kinetic Energy: Assuming uniform linear elongation, the velocity of the microelement $\mathrm{d}x$ at position x along the actuator varies linearly, $\vec{v}(x) = \vec{v}_{\text{A}} \cdot \frac{l-x}{l} + \vec{v}_{\text{B}} \cdot \frac{x}{l}$. By integrating over the entire length, the total kinetic energy of the actuator is given by:

$$T = \int_0^l \frac{1}{2} v^2(x) \frac{m}{l}\, \mathrm{d}x = \frac{1}{6} m (\vec{v}_{\text{A}} \cdot \vec{v}_{\text{B}} + v_{\text{A}}^2 + v_{\text{B}}^2) \tag{4}$$

$$\hat{T} = \sum_{i=1}^{N} \frac{1}{2} m_i \|(1-\xi_i)\vec{v}_{\text{A}} + \xi_i \vec{v}_{\text{B}}\|^2. \tag{5}$$

Let $\mu_i = \frac{m_i}{m}$. Energy-equivalence requires $\hat{V}_{\text{g}} = V_{\text{g}}$, $\hat{T} = T$. By substituting into (2), (3), (4) and (5),

$$\left.\begin{aligned} \hat{T} &= T \\ \hat{V}_{\text{g}} &= V_{\text{g}} \end{aligned}\right\} \Longleftrightarrow \begin{bmatrix} 1 & \cdots & 1 \\ \xi_1 & \cdots & \xi_N \\ \xi_1^2 & \cdots & \xi_N^2 \end{bmatrix} \cdot \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/3 \end{bmatrix}. \tag{6}$$

The linear system admits solutions only if $N \geq 3$. When $N = 3$, the coefficient matrix reduces to a convertible 3rd-order Vandermonde matrix for any (ξ_1, ξ_2, ξ_3) that meets $\xi_1 \neq \xi_2 \neq \xi_3$, ensuring the existence of a solution. That is the necessary condition of a rigid discrete equivalent model.

Without loss of generality, we set $\xi_1 = 0$ and $\xi_3 = 1$, to anchor the ends of the actuator to the skeleton. To achieve better symmetry and a simpler constraint of length, we select $\xi_2 = \frac{1}{2}$, resulting in the solution $\mu_1 = \frac{1}{6}$, $\mu_2 = \frac{2}{3}$, and $\mu_3 = \frac{1}{6}$. This solution corresponds to (7) and (8). Additionally, the midpoint is the optimal choice for interpolation accuracy according to Simpson's interpolation theory.

$$V_{\text{g}} = m g \vec{k} \cdot \left[\frac{1}{6}\vec{r}_{\text{A}} + \frac{2}{3}\left(\frac{\vec{r}_{\text{A}} + \vec{r}_{\text{B}}}{2}\right) + \frac{1}{6}\vec{r}_{\text{B}}\right] = \hat{V}_{\text{g}} \tag{7}$$

$$T = \frac{1}{2} m \left[\frac{1}{6} v_{\text{A}}^2 + \frac{2}{3}\left(\frac{\vec{v}_{\text{A}} + \vec{v}_{\text{B}}}{2}\right)^2 + \frac{1}{6} v_{\text{B}}^2\right] = \hat{T} \tag{8}$$

3) Generalized Forces and Elastic Potential Energy: Under the above assumptions, and by analogy with series rules for springs and dampers, dynamic equivalence holds if the following

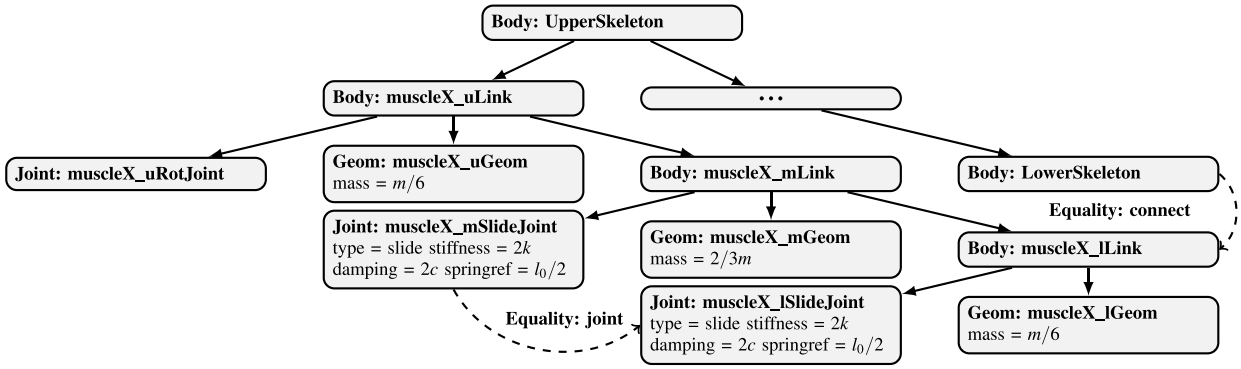


Fig. 3. MJCF hierarchical structure of the EquiMus model, showing body–joint–geom relationships and key attributes. The "..." node denotes the remaining rigid skeleton structure, omitted here for clarity. Dashed arrows indicate <equality> constraints, including joint equality and body connection.

conditions are met:

$$\begin{cases} F_1 = F_2 = F, & c_1 = c_2 = 2c, \\ k_1 = k_2 = 2k, & l_{10} = l_{20} = l_0/2. \end{cases} \tag{9}$$

The elastic potential energy and virtual work brought by generalized forces (driving force F and the damping force $c\dot{l}$) matches:

$$\begin{aligned} \hat{V}_\mathrm{e} &= \frac{1}{2}k_1(l/2 - l_{10})^2 + \frac{1}{2}k_2(l/2 - l_{20})^2 \\ &= 2 \times \frac{1}{2} \times 2k\left[(l - l_0)/2\right]^2 = \frac{1}{2}k\left[(l - l_0)\right]^2 = V_\mathrm{e} \end{aligned} \tag{10}$$

$$\begin{aligned} \delta\hat{W} &= (F_1 - c_1\dot{l}/2)\vec{e}_{\mathrm{CA}} \cdot \delta\vec{r}_{\mathrm{CA}} + (F_2 - c_2\dot{l}/2)\vec{e}_{\mathrm{CB}} \cdot \delta\vec{r}_{\mathrm{CB}} \\ &= 2 \times (F - c\dot{l})\vec{e}_{\mathrm{BA}} \cdot 1/2\delta\vec{r}_{\mathrm{BA}} = \delta W \end{aligned} \tag{11}$$

where $\vec{e}_{\mathrm{AB}}$ denotes the unit vector from A to B. According to the principle of virtual work, the equivalence in virtual work implies that of generalized forces. Consequently, the transformation constructed above preserves all physical terms, resulting in full equivalence of the system dynamics.

D. Summary

Shown in Fig. 2, the dynamic model of the linear elastic actuator can be equivalently represented by a discrete mass system. The method follows a 3-2-1 approach:

- **3 Mass Points**: The actuator is discretized into three mass points—two fixed at each end and one at the midpoint—with $\frac{1}{6}m$, $\frac{1}{6}m$, and $\frac{2}{3}m$ respectively.
- **2 Linear Actuators**: Two motors connect the masses, each with stiffness $2k$, damping coefficient $2c$, rest length of $l_0/2$, and driving forces F.
- **1 Constraint**: An equality constraint is implicitly enforced through the above conditions, ensuring that the elongation of both actuator segments remain identical.

Our method achieves a balanced trade-off between physical fidelity and computational efficiency. On one hand, the equivalent model is rigorously constructed based on energy equivalence, carefully accounting for the variable mass distribution, thereby preserving high accuracy. On the other hand, although the soft actuator is discretized into a multi-rigid-body representation, the number of equivalent elements and constraints remains comparable to that of the original actuator, substantially reducing computational overhead.

TABLE II
ACTUATOR PROPERTIES

Actuator	Mass (kg)	Len. (m)	Damp. (Ns/m)	Stiff. (N/m)	Area (m^2)
MAA	0.1865	0.1744	10.8	367.8	6.54×10^{-4}
BAA	0.2727	0.2536	11.3	291.8	6.37×10^{-4}

E. Software Implementation

We choose MuJoCo [15] as simulation platform due to its effective support for kinematic loops. The energy-equivalent implementation of the linear elastic actuator is illustrated in Fig. 3. Parameters are configured with a 1:4:1 <mass> distribution, double <stiffness> and <damping> coefficients, and half <springref> (rest length). An <equality> on joint constraint maintains equal actuator lengths, keeping the middle mass point fixed at the geometric midpoint.

IV. EXPERIMENTS AND RESULTS

A. Experimental Platform of Rigid–Soft Hybrid Robotic Leg

1) Overview of System: We validate EquiMus on a pneumatic rigid–soft robotic leg [9]. The leg, shown in Fig. 4, consists of a base, hip joint, thigh, knee joint, calf, a mono-joint actuator (MAA), and a bi-joint actuator (BAA). Rigid links are 3D printed from nylon, and linear elastic actuators are cast from polyurethane rubber. The overall system includes:

- **Actuation:** Two pneumatic artificial muscles (PAMs) are driven by proportional pressure valves (VPPE-3-1/8-6-010, Festo). Each actuator is commanded in the range of 0–50 kPa [9] (air supply up to 600 kPa).
- **Sensoring:** Joint motion is tracked by a motion-capture system (Optitrack, Natural Point) at 120 Hz.
- **Control:** A ROS2-based PC receives state estimates and outputs valve pressure commands via DACs.

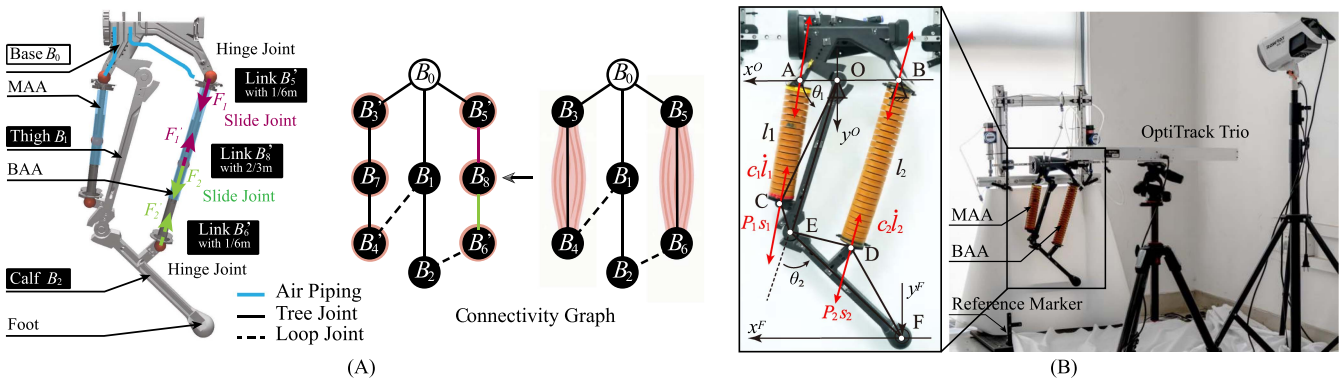


Fig. 4. Comparison between simulation and physical implementation of the robotic leg system. (A) MuJoCo-based energy-equivalent model of the robotic leg, illustrating the complete structure and the corresponding connectivity graph. Black nodes represent rigid links; black solid lines indicate tree joints; black dashed lines denote loop-closure joints (loop joints). Two blue lines represent pneumatic tubing. For clarity, the MAA module is omitted, and the focus is on the BAA structure. (B) Physical experimental setup, including the coordinate frame of the bionic robotic leg with pneumatic artificial muscles and the OptiTrack motion capture system. A simplified force analysis diagram is also included to aid understanding of the joint loading conditions.

The actuators' physical parameters are summarized in Table II .

2) Kinematic Model of Robotic Leg: We consider two phases: swing (base fixed) and stance (foot in contact). Inputs are actuator pressures $\mathbf{P} = [P_1, P_2]$, which generate axial forces $\mathbf{F} = [F_1, F_2] = [S_1 P_1, S_2 P_2]$ via the effective cross-sectional areas S_i. The outputs are joint angles $\mathbf{q} = [\theta_1, \theta_2]$ of the hip and knee.

For swing phase, a coordinate system is established as shown in Fig. 4(b), with the origin O located at the hip joint. The positions and velocities of all key points, O, A, ... H, as well as the lengths of the two actuators, l_1, l_2, can be computed analytically through forward kinematics.

For the stance phase, by choosing the foot position as the origin of the reference frame, the kinematic model can be reformulated via a coordinate transformation $\mathbf{r}^{\mathrm{F}} = \mathbf{r}^{\mathrm{O}} - \overrightarrow{\mathrm{OF}}$.

B. Implementation of Dynamic Simulations

1) Theoretical Dynamic Model and Simulation: Based on (2), (4), (10), and (11), we derived the dynamics of the robotic leg via the Euler–Lagrange equation, obtaining $\dot{\mathbf{q}} = f_{\text{theo}}(\mathbf{q}, t)$. Numerical simulation was implemented in MATLAB/SymPy using the `ode45` solver, which serves as the theoretical ground truth.

2) Software Pipeline: The topology transformation and the schematic diagram of the mechanism are shown in Fig. 4. We implemented the equivalent model of the robotic foot through a systematic process. A URDF file (with loop closures opened) was converted to MJCF, and the 3-2-1 structure (Section III-D) was added with the necessary bodies/joints and constraints. Equality constraints were introduced to enforce the actuator lengths and their connections to the skeleton. The simulator runs at 141.9× real time (mean wall-clock step 0.0071 ms; $\Delta t = 0.005$ s), sufficient for real-time control and repeated simulation experiments.

TABLE III
ERROR COMPARISON BETWEEN THEORETICAL AND EQUIVALENT MODELS, IN STATIC AND DYNAMIC TESTS

Metric	Static θ_1 / θ_2	Dynamic-Swing θ_1 / θ_2	Dynamic-Stance θ_1 / θ_2
RMSE (rad)	0.00096 / 0.00574	0.00094 / 0.00586	0.01693 / 0.03000
MaxAE (rad)	0.00369 / 0.01452	0.00379 / 0.01559	0.04673 / 0.08805

C. Verification in Simulation

1) Static Equivalence: For each static state $\mathbf{q}$ sampled in the workspace $\Omega = \{(\theta_1, \theta_2) \mid \theta_1 \in [\pi/6, 2\pi/3], \theta_2 \in [0, \pi/2]\}$ we compared equilibrium configurations of the theoretical and equivalent models under identical balancing force inputs $\mathbf{F} = f(\mathbf{q})$. Both models shared the same parameters $(l_{10}, l_{20}, k_1, k_2)$ with small perturbations, and $c_1 = c_2 \approx 10\,\mathrm{Ns/m}$. Table III summarizes errors over 10,000 trials: RMSEs (root-mean-square error) for θ_1 is < 0.001 rad and for θ_2 is < 0.06 rad, indicating close static agreement.

2) Dynamic Equivalence:

2) Swing Phase: We uniformly sampled $(\mathbf{q}, \mathbf{q}') \in \Omega \times \Omega$, computed $\mathbf{F} = f(\mathbf{q})$ and $\mathbf{F}' = f(\mathbf{q}')$, and finally recorded the step response from the static state with $\mathbf{F}$ to that of $\mathbf{F}'$. Of 10,000 trials, 2,407 valid trajectories remained after discarding joint-limit violations.

2) Stance Phase: To test the generalization of our method, we change the working condition to stance phase. The init state is chosen as standing still $(\theta_1 = \pi/2, \theta_2 = 0)$, the impulse force $\mathbf{F}'$ as $(10\,\mathrm{N}, 10\,\mathrm{N})$.

The errors of both phases are summarized in Table III, and trajectories of several experiments are visualized in Fig. 5. The max absolute error (MaxAE) of all experiments of swing phase is less than 0.016 rad (~ 1 deg). The error of stance phase is about 5 times larger, which is still a small value compared to the workspace. The larger error may be caused by the relatively weak constraint between the foot and the ground. The results show that the equivalent model exhibits a small deviation from the

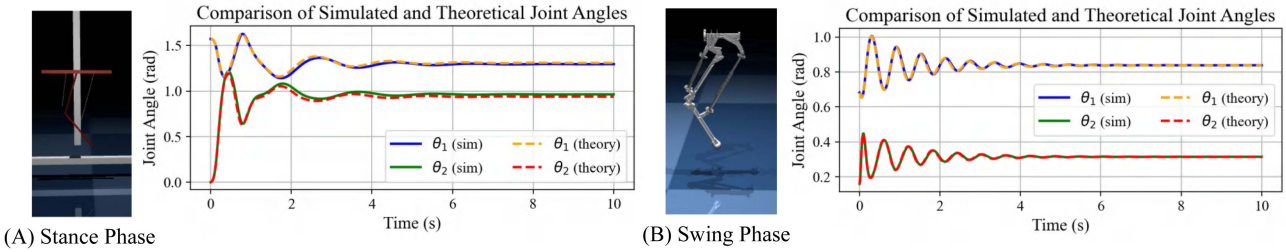


Fig. 5. Verification of dynamic equivalence in simulation. The stance phase (A) and swing phase (B) are tested. Joint trajectories from simulation and theoretical models show strong agreement, demonstrating the validity of the proposed formulation.

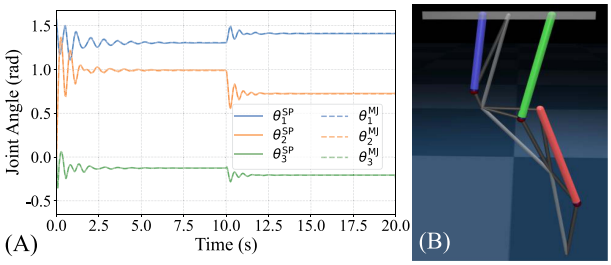


Fig. 6. Morphology generalization on a 3-DOF musculoskeletal robot. (A) Joint-angle trajectories under a multi-pulse actuation sequence; solid: analytical (SymPy), dashed: EquiMus (MuJoCo). (B) Randomized 3-DOF morphology with three muscles highlighted.

theoretical model in step response, confirming the effectiveness and accuracy of the proposed equivalent modeling methods.

3) Morphology Generalization: We evaluated EquiMus on a randomized 3-DOF musculoskeletal robotic leg, varying muscle routing and joint-connection topology. Under identical inputs, time step, and solver settings, we applied a multi-pulse actuation sequence and compared joint trajectories from the analytical model (SymPy) against those from EquiMus, obtaining per-joint RMSEs of 0.0035, 0.0066, and 0.0041 rad, as illustrated in Fig. 6. Those small errors validate that the energy-equivalent unit generalizes well to novel topologies, supporting the structural generalization of the framework.

D. Physical Validation

To make comparison of the simulation with real systems, parameter identification is conducted at both the actuator level and the system level, aiming to capture the key characteristics of individual actuators as well as the global static and dynamic behavior of the robotic leg.

1) Actuator Calibration: Actuator-level calibration was performed to determine parameters such as m and l_0.

2) Static Parameter Identification and Verification: We tested the steady-state response with specific pressure commands. The data was recorded in the form $(\theta_1, \theta_2, P_1, P_2)$, representing the joint angles and corresponding actuator pressures at equilibrium. We used the theoretical static model to calibrate static parameters, then calibration on (k_1, k_2, s_1, s_2) became a linear regression problem.

In the experiment, the pressure was selected from 0–50 kPa at intervals of 10 kPa, totaling 36 experiments (Set B). After excluding the experiments with antagonistic situations, 13 configurations $[P_1, P_2, \theta_1, \theta_2]$ are remained, forming Set A, which is used for system-level parameter identification. The results are listed in Tab. V. The RMSE of pressure is 0.887 kPa for P_1 and 0.861 kPa for P_2, both below 2% of the maximum applied pressure (50 kPa), indicating a reliable regression accuracy.

TABLE V
COMPARISON OF STATICALLY AND DYNAMICALLY IDENTIFIED PARAMETERS

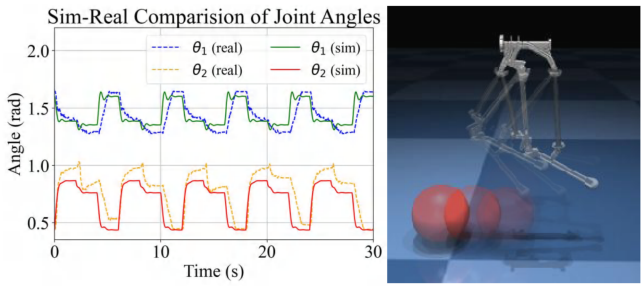
Parameter	Static Value	Dynamic Value
k_1 (N/m)	203.95	265.82
k_2 (N/m)	105.07	110.36
s_1 (m^2)	0.000411	0.000403
s_2 (m^2)	0.000324	0.000436
l_{10} (m)	0.1642	0.1803
l_{20} (m)	0.2579	0.2565
c_1 (Ns/m)	–	12.48
c_2 (Ns/m)	–	24.79
c_{hip} (Ns/m)	–	0.524
c_{knee} (Ns/m)	–	0.062

TABLE IV
STATIC VALIDATION RESULTS ACROSS TWO EVALUATION SETS

Evaluation Set	RMSE θ_1	RMSE θ_2	MaxAE θ_1	MaxAE θ_2
Set A (13 poses)	0.0170	0.0566	0.0336	0.1463
Set B (36 poses)	0.0203	0.0416	0.0418	0.1463

The parameters identified from Set A were then used to simulate the same pressure inputs for both configurations of Set A and Set B. The static joint angles were compared against experimental measurements. For Set A, compared to the actual positions, only one data point had an error exceeding 0.087 rad. This verifies the accuracy of the static model. As reported in Table IV, the model maintains low angular errors in both Set A and B. This confirms that our energy-equivalent modeling framework generalizes well, even in highly nonlinear and boundary-range conditions. Notably, larger errors in θ_2 tend to occur when θ_1/θ_2 is near its joint limit. For example, MaxAE of θ_2 in Set A and Set B occur simultaneously at $\theta_{1,2} = 2.068, 0.18$ rad, where θ_1 is very close to its upper bound $2/3\,\pi$. In such poses, nonlinear effects caused by static friction and boundary constraints become more significant, making accurate simulation more challenging. While this introduces slight discrepancies, the overall predictive performance remains robust and suitable for downstream applications such as control and planning.

3) Dynamic Calibration and Verification: We tested step response in real world experiments and simulations to calibrate the main dynamic parameters $\mathbf{c} = [k_1, k_2, s_1, s_2, l_{10}, l_{20}, c_1, c_2, c_{\text{hip}}, c_{\text{knee}}]$. We tracked the actual



(A) Open Loop Control Command Comparison (B) Ball Kicking with Learning

Fig. 7. (A) Tracking performance of the triangular trajectory in both simulation and hardware experiments. Each vertex corresponds to a pressure command $P_{1,2}$: (1) 6.15, 2.73 kPa; (2) 3.76,7.26 kPa; (3) 19.80,8.52 kPa. (B) The robotic leg learns to kick a ball based on reinforcement learning.

joint trajectory $\tau_i(t)$ and the simulated trajectory $\hat{\tau}_i(t, \mathbf{c})$, respectively. We continued to use 13 groups got in last experiments. Finally we wanted to minimize the position error with the interpolation form of trajectory:

$$\mathbf{c} = \arg\min \sum_{i=1}^{13} E(\boldsymbol{\tau}_i, \hat{\boldsymbol{\tau}}_i(\mathbf{c})). \tag{12}$$

Define the error $E(\theta_1, \theta_2) = \text{MSE}(\tau(t)) + \lambda \text{MSE}(\text{sgn}(\dot{\tau}(t)))$ Here, λ (set as 100in experiments) is the weighting parameter that penalizes angular velocity signal error. The error represents the deviation in the system's motion state $[\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]$.

Our work employed differential evolution algorithm (DE) to do optimizations, with up to 10,000 iterations, and a population size of 25. Final parameters are shown in Table V. Most of the parameters calibrated from the dynamic identification is closer to that calibrated from the static identification. The RMSE of the angle is under 0.03253 rad for all experiments. It indicates the validness of the equivalent method. The error may because of the nonlinearity of the actuator, which carry in some bias on the actuator dynamics.

In both MuJoCo simulation and real-world experiments, we commanded the robot to follow a triangular trajectory using a sequence of pressure steps. The resulting joint trajectories are shown in Fig. 7. They exhibit similar dynamic characteristics. The RMSE of θ_1 and θ_2 are 0.092 and 0.174 rad respectively. Some discrepancies between simulated and actual joint trajectories, especially in θ_2, arise due to the joint's higher sensitivity to system disturbances. The presence of static friction may cause sticking or abrupt transitions and the effect of control latency in pneumatic valves may introduces slight time mismatches.

4) Baseline Comparison With Native MuJoCo Implementation: To benchmark the effectiveness of our energy-equivalent formulation, we constructed a baseline using MuJoCo's native motor actuators configured as single spring-damper elements between bodies, without internal mass or constraint enforcement. The simulation was run under the same pressure inputs and mechanical boundary conditions as in our full model. As shown in Table VI, the baseline's accuracy deteriorates significantly. These results underscore the necessity of incorporating intermediate mass elements and equality constraints to more accurately capture the actuator's physical behavior. This comparison highlights the advantages of our approach in preserving both simulation fidelity and real-time performance, making it well-suited for reinforcement learning and control applications.

TABLE VI
BASELINE COMPARISON BETWEEN EQUIMUS AND NATIVE MUJOCO

Method	RMSE (rad)	Notes
EquiMus	0.032522	Energy-equivalent formulation
PureMuJoCo	0.432144	Native actuator, lumped model

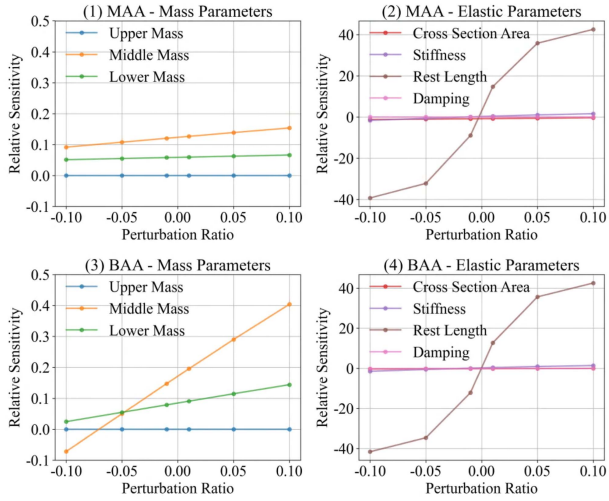


Fig. 8. Sensitivity analysis of the model parameters.

E. Sensitivity Analysis of Model Parameters

To evaluate robustness under parametric uncertainty, we perturbed key actuator parameters (m, k, l, c, s) by $\pm 1\%$, $\pm 5\%$, $\pm 10\%$ and measured the change in joint-angle RMSE with respect to real-world data. As shown in Fig. 8, the model remains robust under small perturbations, though rest lengths (l_{10}, l_{20}) and lower-link masses exhibit higher sensitivity. These critical parameters were precisely calibrated to ensure model fidelity during implementation.

F. Application-Driven Demonstrations

Built on MuJoCo, EquiMus leverages that ecosystem's XML tooling, Gym-compatible interfaces, and mainstream RL/control libraries, enabling direct integration into standard pipelines with minimal additional code. We illustrated three representative usages. (i) **Reinforcement learning**: A ball-kicking task was achieved with PPO (Stable-Baselines3), with results shown in Fig. 7 and the video. (ii) **Parameter exploration/identification**: Fast batched simulation supports sweeps of parameter like stiffness for system design and identification. (iii) **Model-based control**: Identified dynamics are deployable within MPC frameworks [39]. We also provide results of an auto PID tuning experiment at https://github.com/fly-pigTH/EquiMus/tree/main/src/application/PID_AutoTuning/ReadMe.md. These demonstrations indicate that EquiMus is not only physically consistent

with hardware but also operationally practical for learning and control.

V. Conclusion

This letter proposed EquiMus, an energy-equivalent dynamics and simulation for the rigid–soft musculoskeletal robots with linear elastic actuators. By lumping actuator inertia with a 3–2–1 discretization, the method captures dynamic mass redistribution, supports loop-closure constraints in MuJoCo, and remains real-time capable. Experiments on a pneumatic leg show close sim-to-real agreement and enable downstream usage in PID auto-tuning, model-based control, and reinforcement learning. A current limitation is the omission of internal actuator nonlinearities; the modular energy-based structure readily admits richer constitutive models as future work. We hope EquiMus provides a novel perspective and serves as a small step toward bridging the gap between soft robotics and embodied intelligence.

References

[1] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: A bioinspired evolution in robotics," *Trends Biotechnol.*, vol. 31, no. 5, pp. 287–294, 2013.

[2] F. Iida and A. J. Ijspeert, "Biologically inspired robotics," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016, pp. 2015–2034.

[3] W. R. Roderick, M. R. Cutkosky, and D. Lentink, "Bird-inspired dynamic grasping and perching in arboreal environments," *Sci. Robot.*, vol. 6, no. 61, 2021, Art. no. eabj7562.

[4] Y. Chen et al., "A biologically inspired, flapping-wing, hybrid aerial-aquatic microrobot," *Sci. Robot.*, vol. 2, no. 11, 2017, Art. no. eaao5619.

[5] J. Hughes, P. Maiolino, and F. Iida, "An anthropomorphic soft skeleton hand exploiting conditional models for piano playing," *Sci. Robot.*, vol. 3, no. 25, 2018, Art. no. eaau3098.

[6] Y. Feng et al., "Experimental validation of a 7-DOF power soft robot driven by hydraulic artificial muscles," *IEEE Robot. Automat. Lett.*, vol. 9, no. 6, pp. 5472–5479, Jun. 2024.

[7] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *Int. J. Robot. Res.*, vol. 35, no. 1-3, pp. 161–185, 2016.

[8] M. T. Tolley et al., "A resilient, untethered soft robot," *Soft Robot.*, vol. 1, no. 3, pp. 213–223, 2014.

[9] X. Dong et al., "Design and control of a musculoskeletal bionic leg with optimized and sensorized soft artificial muscles," *IEEE Trans. Robot.*, vol. 41, pp. 3402–3422, 2025.

[10] X. Zhang, F. K. Chan, T. Parthasarathy, and M. Gazzola, "Modeling and simulation of complex dynamic musculoskeletal architectures," *Nature Commun.*, vol. 10, no. 1, 2019, Art. no. 4825.

[11] C. D. 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, Jun. 2023.

[12] M. A. Graule, C. B. Teeple, T. P. McCarthy, G. R. Kim, R. C. S. Louis, and R. J. Wood, "Somo: Fast and accurate simulations of continuum robots in complex environments," in *Proc. 2021 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 3934–3941.

[13] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[14] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016–2021. [Online]. Available: http://pybullet.org

[15] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[16] V. Makoviychuk et al., "ISAAC GYM: High Performance GPU Based Physics Simulation For Robot Learning," *NeurIPS Datasets Benchmarks*, 2021.

[17] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, 2022, Art. no. eabk2822.

[18] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. Conf. Robot Learn.*, 2022, pp. 91–100.

[19] Z. Li et al., "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 2811–2817.

[20] H. Wang, V. Caggiano, G. Durandau, M. Sartori, and V. Kumar, "Myosim: Fast and physiologically realistic mujoco models for musculoskeletal and exoskeletal studies," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 8104–8111.

[21] V. Caggiano, H. Wang, G. Durandau, M. Sartori, and V. Kumar, "MyoSuite: A contact-rich simulation suite for musculoskeletal motor control," in *Proc. Learn. Dyn. Control Conf.*, 2022, pp. 492–507.

[22] 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, Jun. 2023.

[23] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 723–746, 2019.

[24] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, 2010.

[25] H. Habibi et al., "A lumped-mass model for large deformation continuum surfaces actuated by continuum robotic arms," *J. Mechanisms Robot.*, vol. 12, no. 1, 2020, Art. no. 0 11014.

[26] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and Jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 823–834, Aug. 2015.

[27] M. A. Graule, T. P. McCarthy, C. B. Teeple, J. Werfel, and R. J. Wood, "Somogym: A toolkit for developing and evaluating controllers and reinforcement learning algorithms for soft robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4071–4078, Apr. 2022.

[28] M. Chignoli, J.-J. Slotine, P. M. Wensing, and S. Kim, "Urdf : An enhanced urdf for robots with kinematic loops," in *Proc. IEEE-RAS 23rd Int. Conf. Humanoid Robots (Humanoids)*, 2024, pp. 197–204.

[29] M. Kasaei, H. Kasaei, and M. Khadem, "Softmanisim: A fast simulation framework for multi-segment continuum manipulators tailored for robot learning," in *Proc. 8th Annu. Conf. Robot Learn.*, 2025, pp. 1473–1500.

[30] R. Jitosho, T. G. W. Lum, A. Okamura, and K. Liu, "Reinforcement learning enables real-time planning and control of agile maneuvers for soft robot arms," in *Proc. Conf. Robot Learn.*, 2023, pp. 1131–1153.

[31] F. Faure et al., "Sofa: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Model. for Comput. Assist. Surg.*. Berlin, Germany: Springer, 2012, pp. 283–321.

[32] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *Int. J. Robot. Res.*, vol. 44, 2024, Art. no. 02783649241285161.

[33] O. M. Andrychowicz et al., "Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[34] S. L. Delp et al., "Opensim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 11, pp. 1940–1950, Nov. 2007.

[35] A. Seth, M. Sherman, J. A. Reinbolt, and S. L. Delp, "Opensim: A musculoskeletal modeling and simulation framework for in silico investigations and exchange," *Procedia Iutam*, vol. 2, pp. 212–232, 2011.

[36] A. Seth et al., "Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement," *PLoS Comput. Biol.*, vol. 14, no. 7, 2018, Art. no. e1006223.

[37] D. Reynolds, D. Repperger, C. Phillips, and G. Bandry, "Modeling the dynamic characteristics of pneumatic muscle," *Ann. Biomed. Eng.*, vol. 31, pp. 310–317, 2003.

[38] T. Mizuno, N. Tsujiuchi, T. Koizumi, Y. Nakamura, and M. Sugiura, "Spring-damper model and articulation control of pneumatic artificial muscle actuators," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2011, pp. 1267–1272.

[39] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," Dec. 2022, *arXiv:2212.00541*.