

A Validated Physical Model For Real-Time Simulation of Soft Robotic Snakes

Renato Gasoto^{*1,2} Miles Macklin^{*2,3} Xuan Liu¹ Yinan Sun¹ Kenny Erleben³ Cagdas Onal¹ Jie Fu¹

Abstract—In this work we present a framework that is capable of accurately representing soft robotic actuators in a multiphysics environment in real-time. We propose a constraint-based dynamics model of a 1-dimensional pneumatic soft actuator that accounts for internal pressure forces, as well as the effect of actuator latency and damping under inflation and deflation and demonstrate its accuracy a full soft robotic snake with the composition of multiple 1D actuators. We verify our model’s accuracy in static deformation and dynamic locomotion open-loop control experiments. To achieve real-time performance we leverage the parallel computation power of GPUs to allow interactive control and feedback.

I. INTRODUCTION

Soft robotic systems can provide many advantages over rigid robots. For example, the compliance of a soft manipulator allows safe interaction with fragile objects, such as uncooked eggs [1, 2]. A soft robotic snake can navigate through narrow passages to perform tasks in regions unreachable with rigid robots [3]. On the other hand, soft robots are harder to model and simulate than rigid robots, due to their infinite degrees of freedom (DOFs). Thus, one often has to compromise between simulation accuracy and simulation time. In recent work [4, 5], the authors developed a framework of deformable online simulation that can achieve real-time simulation using relatively coarse meshes. Further, the authors proposed a model order reduction method by running expensive offline simulations and applying machine-learning techniques on the generated dataset [6].

However, the compromise between accuracy and simulation time is often hard to make. An accurate model is important to bridge the gap between simulation and reality for control design. On the other hand, online trajectory planning and model predictive control require the prediction of future states with low-latency. In addition, even learning-based controllers, such as those trained with Reinforcement Learning (RL) [7, 8, 9], typically require thousands of samples to converge, making efficient simulation crucial. Approximate models can enable faster simulations; Furthermore, even in rigid-body simulators, the modeling errors, inaccuracies, and

lack of actuator dynamics on the simulator create a gap to the reality that often needs to be bridged before applying controllers learned in simulation to the real robot [10].

In this work, we aim to develop a real-time, high-fidelity simulation for soft robotic systems. We present a compliant constraint dynamics model for a soft snake robot [11, 12, 13]. Our model is able to accurately represent the deformations of the snake links while being efficient enough for real-time simulations. To achieve low-latency simulation, we use a GPU-based physics simulator that leverages large-scale parallel iterative solvers to efficiently solve large systems [14]. The resulting simulation is validated against a real robotic snake to verify that the deformation model is accurate and that the dynamics of the simulation match that of the real system. In addition, we measure and validate the actuator and inflation latency to ensure accurate control response. In summary, our main contributions are:

- A dynamics and actuation framework for 1D pneumatic soft actuators that accurately represents a large range of deformations;
- A model for a modular soft robotic snake that accurately represents its dynamics;
- Model validation in static deformation and dynamic locomotion tests;
- A simulator framework suitable for performing real-time control of soft robots.

II. RELATED WORK

The field of soft robots includes a large and varied range of designs that incorporate compliant materials and actuators in various forms. Soft robots may have a few flexible regions in them and may be driven by tendons [15], while completely soft pneumatic actuators can be driven by exerting a range of pressures within deformable bodies. The use of pressure leads to many intricate designs that exploit the material geometry to achieve the desired actuation [1, 16]. To allow a large range of pressures and material deformations, hybrid materials are often used on the pressure chambers, such as inextensible layers and fiber reinforcements [17]. Our modular soft snake robots use hybrid materials [12, 13] to achieve highly efficient 2D terrain locomotion.

The diversity of soft materials and difficulty in accurate modeling of soft robots lead to an increasing need for building a realistic simulation for deformable robots. Duriez et al. [18] presented a framework for simulating soft robots using a quasi-static approach based on Finite Elements (FEM). In

¹ Renato Gasoto, Xuan Liu, and Yinan Sun are PhD students under the supervision of Jie Fu and Cagdas Onal in the Robotics Engineering program at Worcester Polytechnic Institute. ²Renato Gasoto and Miles Macklin are affiliated with NVIDIA. ³Miles Macklin is a PhD student under the supervision of Kenny Erleben at the University of Copenhagen. This work was supported in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant 208053/2014-0, by the National Science Foundation under grant #1728412, and by NVIDIA. ^{*}Authors have equal contribution in paper.

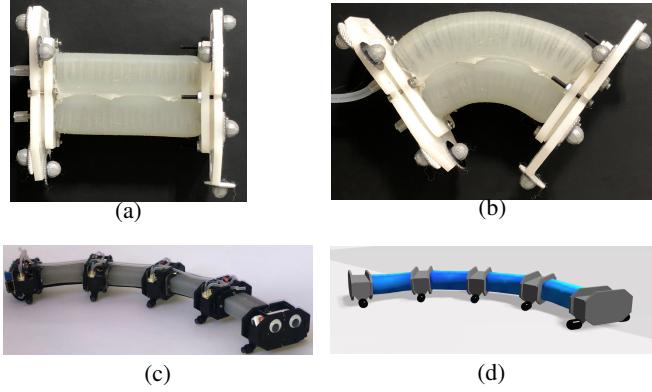


Fig. 1. (a,b) Single soft link. The chambers are wrapped with a fiber reinforcement, preventing it from increasing in radius when pressure is applied. The center of the link contains an inextensible layer that prevents it from expanding in length [11]. The tiny spheres attached to the rigid plates are markers used for tracking the curvature in the experiments. (a) No pressure applied. (b) 8 psi applied on left chamber. (c) Full assembly of the robotic snake with four links. The main controller receives wireless commands from the computer, and passes it over to each slave controller, which activates the solenoid valves that release the pressure to the soft actuator. (d) Snake in simulation.

this work, we also use FEM with tetrahedral elements as a fundamental building-block. In addition, we combine these elements in a multiphysics system with spring networks, frictional contact, attachment constraints for soft/hybrid materials and articulated rigid bodies. We perform implicit time-integration to simulate dynamic trajectories. Recent work by Pozzi et al. [19] used a rigid-body model, fitted to an offline FEM simulation augmented with stiff springs to achieve real-time updates. In this work, we simulate the finite-element models and spring networks directly, using the large-scale parallelism of graphics processing units (GPUs) to achieve online update rates, therefore not requiring an expensive offline simulation and data fitting stage.

Tan et al. [20] propose several techniques for bridging the reality gap in simulations. Among them there is a latency introduced for the delay between when a command is sent and when it's executed. Since soft actuators have an over-damped response, our work introduces system dynamic model response to the command on top of the latency.

III. SOFT ROBOTIC SNAKE

The snake is made of soft bending actuation modules with integrated curvature sensing [11], as shown in Fig. 1c. Each soft bending actuator segment is comprised of two soft linear actuators and an inextensible constraint layer between them. The soft links are made of silicone rubber with cavities wrapped in a fiber reinforcement, which limits the radial deformation when pressurized. In the center of the link, between the two chambers, there is a custom integrated curvature sensor, and a plastic film that inhibits linear extension. This set of constraints results in bending the entire soft module when one chamber is pressurized, as seen in Fig. 1b. The curvature sensor utilizes a magnet and a Hall effect sensor, mounted on a flexible circuit board [11]. Caps

are attached to both ends of the actuator to seal the chambers and allow modular connections with other segments. The caps are made of two ABS plates sandwiching the rim of the silicone. The actuator is driven by two 3-2 (3-port, 2-state) binary solenoid valves, each connecting one pressure chamber to a common pressure source. The valves can either inflate or deflate a given actuator chamber. The pressure in each chamber is controlled using a 60 Hz Pulse-Width Modulation (PWM) on the valves, which are set to operate in complete antagonism, so when one chamber is inflating the other is always deflating. This reduces the number of required inputs per chamber to one, corresponding to the single (active) DOF of the bending actuator.

IV. SIMULATOR

The continuous equations of motion for the multiphysics simulator are derived from Lagrangian mechanics, and are given in general form by the following,

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} - \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{J}_b^T \boldsymbol{\lambda}_b - \mathbf{J}_n^T \boldsymbol{\lambda}_n - \mathbf{J}_f^T \boldsymbol{\lambda}_f &= \mathbf{0} \\ \mathbf{c}_b(\mathbf{q}, \mathbf{p}) + \mathbf{E}\boldsymbol{\lambda}_b &= \mathbf{0} \\ \mathbf{0} \leq \mathbf{c}_n(\mathbf{q}) \perp \boldsymbol{\lambda}_n \geq \mathbf{0} \\ \forall i \in \mathcal{A}, \quad \mathbf{D}_i^T \dot{\mathbf{q}} + \frac{|\mathbf{D}_i^T \dot{\mathbf{q}}|}{|\boldsymbol{\lambda}_{f,i}|} \boldsymbol{\lambda}_{f,i} &= \mathbf{0} \\ \forall i \in \mathcal{A}, \quad 0 \leq |\mathbf{D}_i^T \dot{\mathbf{q}}| \perp \mu_i \boldsymbol{\lambda}_{n,i} - |\boldsymbol{\lambda}_{f,i}| \geq 0 \\ \forall i \in \mathcal{I}, \quad \boldsymbol{\lambda}_{f,i} &= \mathbf{0}. \end{aligned}$$

These equations describe the motion of a generic dynamics system with frictional contact forces. The state of the system is described by a vector of generalized coordinates $\mathbf{q} \in \mathbb{R}^{n_d}$ with n_d DOFs, determined by the number of particles and rigid bodies on the system. The inertial properties of the system are represented by the mass-matrix $\mathbf{M} \in \mathbb{R}^{n_d \times n_d}$, with $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}})$ a generalized force function that includes external and gyroscopic forces. The vector $\mathbf{c}_b(\mathbf{q})$ is a set of bilateral constraints of length n_b , with $\boldsymbol{\lambda}_b$ the associated Lagrange multipliers. Elastic energy potentials are defined in terms of compliant constraints, here $\mathbf{E} \in \mathbb{R}^{n_b \times n_b}$ is a block-diagonal compliance, or inverse stiffness matrix as described by Servin et al. [21]. The target pressures are grouped into the vector \mathbf{p} , which are parameters to the actuation constraints described in section IV-C. The contact and frictional forces are based on Coulomb's model, which defines an admissible cone of contact forces [22]. Here $\mathbf{c}_n(\mathbf{q})$ are unilateral contact constraints, with n_c the number of contacts in the system, and $\boldsymbol{\lambda}_{n,i}$ and μ_i the normal force Lagrange multiplier and friction coefficient for the i th contact respectively. The frictional forces for a contact are parameterized by $\boldsymbol{\lambda}_{f,i}$, with a corresponding basis \mathbf{D}_i that defines the surface tangent plane at the contact point. The active contact set is defined as $\mathcal{A} = \{i \in (1, \dots, n_c) \mid \mu_i \boldsymbol{\lambda}_{n,i} > 0\}$, with inactive contacts \mathcal{I} being its complement. The constraint Jacobians $\mathbf{J}_b, \mathbf{J}_n$ contain the gradient of bilateral and normal constraint functions with respect to \mathbf{q} , and we define the set of frictional basis vectors as the matrix $\mathbf{J}_f = [\mathbf{D}_1, \dots, \mathbf{D}_{n_c}]^T$.

A. Particles

Each deformable link is modeled as a collection of particles connected by constraints. This is a flexible representation that allows fine-grained control over different sections of the soft body, while being efficient enough for real-time simulation. A particle with index i adds three additional DOFs to the system,

$$\mathbf{q}_i = [x \ y \ z]^T. \quad (1)$$

Assuming a lumped mass model, each particle is assigned a fraction of the connected tetrahedral elements mass (Section IV-C.2). The mass-block for the particle is then given by $\mathbf{M}_i = m\mathbf{1}_3$, where m is the particle mass, and $\mathbf{1}_3$ is the 3-dimensional identity matrix.

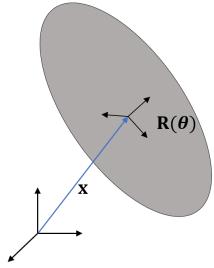


Fig. 2. Rigid links and wheels are described by the translation of the body's center of mass from the origin \mathbf{x} and, it's orientation expressed as a quaternion θ .

B. Rigid Bodies

We describe the state of a rigid body with index i using a maximal coordinate representation consisting of the position of the body's center of mass, $\mathbf{x}_i \in \mathbb{R}^3$, and its orientation expressed as a quaternion $\theta_i = [\theta_1, \theta_2, \theta_3, \theta_4]^T \in \mathbb{R}^4$. We group these components together so the state sub-block for a single rigid body is

$$\mathbf{q}_i = [\mathbf{x}_i^T \ \theta_i^T]^T. \quad (2)$$

C. Constraints

1) *Actuation Constraints:* To perform actuation we constrain particles together through equations of the form,

$$c_{dist}(\mathbf{q}, p) = |\mathbf{q}_i - \mathbf{q}_j| - r\epsilon(p) = 0, \quad (3)$$

where \mathbf{q}_i and \mathbf{q}_j are particle positions, and r is a rest length to maintain between them. The target pressure p induces a strain $\epsilon(p)$ that adjusts the rest length and causes contraction or expansion. Assuming that deformation is linear with stress, and that it occurs primarily along the chamber's main axis, the amount of expansion/contraction is given by the following relation between material stiffness determined by the Young's Modulus (Y) and pressure p ,

$$\epsilon(p) = 1 + p/Y. \quad (4)$$

Furthermore, we use distance constraints with constant rest length to model the structural stiffness in the deformable chamber, as described in Sec. V.

2) *Tetrahedral Finite-Elements:* In addition to distance constraints, tetrahedral finite-elements are used to model the solid chamber material. Assuming a constant strain element and a linear isotropic constitutive model, each tetrahedron defines a 6-dimensional constraint vector,

$$\mathbf{c}_{tetra}(\mathbf{q}) + \mathbf{E}_{tetra}\boldsymbol{\lambda} = \mathbf{0}, \quad (5)$$

where $\mathbf{c}_{tetra}(\mathbf{q}) = [\epsilon_{xx} \ \epsilon_{yy} \ \epsilon_{zz} \ \epsilon_{yz} \ \epsilon_{xz} \ \epsilon_{xy}]^T$ is the vector of corotational strains in Voigt notation, and \mathbf{E}_{tetra} is the constant element compliance matrix, given by

$$\mathbf{E}_{tetra} = \frac{1}{V_e Y} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1+\nu \end{bmatrix}.$$

where V_e is the element volume, Y and ν are the material Young's modulus and Poisson's ratio, respectively.

3) *Rigid Body to Particle Attachment:* In order to connect soft links to rigid bodies, an attachment constraint between a particle and a point on a rigid body is defined as follows,

$$\mathbf{c}_{attach}(\mathbf{q}) = \mathbf{q}_x + \mathbf{R}(\mathbf{q}_\theta)\mathbf{r} - \mathbf{q}_p = \mathbf{0}. \quad (6)$$

This is a vector-valued function that adds three separate constraints, one for each x, y, z axis respectively. Here $\mathbf{q}_x, \mathbf{q}_\theta$ are the rigid body position and orientation respectively, and \mathbf{q}_p is the particle position. $\mathbf{R}(\mathbf{q}_\theta)$ is the rotation matrix obtained from the body's orientation, and \mathbf{r} is the attachment point expressed in the body's local frame.

4) *Rigid Body Joints and Contact:* Along with the deformable sections, we model the articulated carriage as rigid bodies, with wheels connected to the main frame using hinge joints as described by [23]. Contact between the wheels and the ground is modeled by non-interpenetration constraints of the form:

$$\mathbf{c}_n(\mathbf{q}) = \mathbf{n}^T [\mathbf{a}(\mathbf{q}) - \mathbf{b}(\mathbf{q})] \geq 0, \quad (7)$$

where $\mathbf{n} \in \mathbb{R}^3$ is the contact plane normal, \mathbf{a} and \mathbf{b} are points on a rigid or deformable body. Frictional forces are included using a Coulomb model derived from a principle of maximum dissipation that limits the contact forces to a cone. We refer the readers to the survey paper by Stewart [24] for more detail.

D. Time-Stepping

The simulation is advanced in time with a first-order implicit time-discretization of the equations of motion similar to the method in [25]. An implicit discretization is chosen as it allows taking large time-steps and avoids constraint drift. At each time-step, the nonlinear system of equations resulting from the implicit discretization is solved using Newton's method. To solve the complementarity conditions associated with contact we use a non-smooth reformulation based on the Fischer-Burmeister function as described in

[26]. Each Newton iteration requires the solution of a sparse-matrix equation of the form

$$[\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T + \mathbf{E}] \Delta\lambda = \mathbf{b}. \quad (8)$$

Where $\mathbf{J} = [\mathbf{J}_b^T \ \mathbf{J}_n^T \ \mathbf{J}_f^T]^T$ is the matrix of constraint Jacobians, \mathbf{E} is a block-diagonal compliance matrix that includes the tetrahedral compliance matrices, and \mathbf{b} includes the constraint function residuals evaluated at the current Newton iterate. This is a positive semi-definite system that we solve using the Preconditioned Conjugate Residual method (PCR) [27]. This is an iterative Krylov method similar to Conjugate Gradient (CG) but with smooth error reduction, making it better suited for real-time applications with a fixed computational budget. Like CG, the primary computation cost of PCR is the performing sparse matrix-vector multiplications. However, these multiplications are highly parallelizable, and can be done efficiently by assembling \mathbf{J} , \mathbf{M} , \mathbf{E} , \mathbf{b} on the GPU in compressed row-storage (CSR) format, and performing the multiplication with optimized kernels [14]. In our simulator we use a simple diagonal Jacobi preconditioner since it is trivial to parallelize.

V. SOFT ROBOTIC SNAKE MODEL

The soft links of the snake robot are made of Ecoflex™ 00-30 silicone rubber which has material parameters $Y = 66.243\text{KPa}$, and $\nu = 0.4999$ [28]. We construct a triangular mesh of the surface and tetrahedralize it using TetGen [29]. The link mesh was created with evenly distributed particles, we do not explicitly represent the cavity with tetrahedral elements. The mesh was carefully constructed to provide a radially symmetrical tetrahedron structure, as seen in Fig. 3.

Since the inextensible layer in the center of the link has a deformation threshold that is beyond the range of forces to be applied on the soft links, it is acceptable to model it as a non-deformable constraint between particles along the center plane. Similarly, the radial constraint on the chambers are defined as a set of inter-particle constraints over coplanar particles along the link. Although it would be possible to drive each link's expansion using surface pressure forces directly, the other constraints in the link allow us to simply control the chamber volume using constraints between particles along the primary axis of expansion.

Only one chamber on the link is active (i.e. pressurized) at a time, so this set of actuation constraints only applies to the expanding chamber. Fig. 3 displays the constraints overlaid on the link. The link mesh was subdivided in 13 cross-sections along its length, in order to allow real-time computation, while maintaining good accuracy on the material deformation.

The links are then connected to each other through the rigid bodies that contain the electronics necessary to control the snake robot. In addition, the rigid bodies are attached to the wheels via. hinge joints. The wheels provide contact with the floor and model the anisotropic friction that a real snake has from its scales.

Type	Quantity
Rigid Bodies	15
Particles	1504
Distance Constraints	1460
Tetrahedral Finite Elements	4536
Rigid Joints	10
Particle Attachments	217

TABLE I
SIZE OF THE STRUCTURE FOR ONE SIMULATED SNAKE

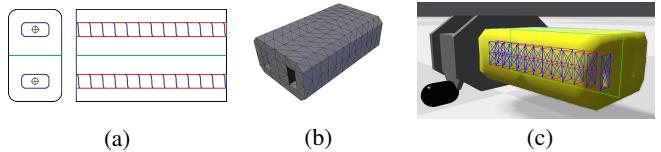


Fig. 3. (a) Front and top view of chamber with constraints between particles on link. Green: stiff inter-particle constraint to limit chamber expansion. Blue: stiff constraints to ensure chamber radial perimeter is constant. Red: distance constraint used to expand chamber as overpressure is applied. (b) Soft link mesh. (c) Constraints displayed on simulation (best seen in digital format).

The type and number of all DOFs, and constraints in the simulated snake are displayed in Table I.

A. Open-Loop Control

The snake assembly consists of four links attached together, as seen in Fig. 1. An undulating motion that propels the snake is given by the control equation 9, which outputs the pressure for the link i .

$$a_i \equiv \min(1, \max(-1, (\sin(\omega t + \alpha i) + \phi)) A. \quad (9)$$

If a_i is positive, the link will inflate one chamber of the link, while if a_i is negative, it will inflate the other. The parameters ω , α , ϕ , and A are the base oscillation frequency, measured in Hz , the phase shift between links (radians), the offset value for the motion ($[-1, 1]$, scalar), and the oscillation amplitude ($[0, p_{\max}]$, Pa), respectively. The oscillation frequency dictates how fast the actuators will switch direction, and the phase delay between links is what generates the wave pattern that propels the snake forward. These parameters set the base undulating motion. By changing the offset ϕ , applied to all links, the controller will inflate one side for longer than the other. This results in the snake propelling itself more to the opposite direction than the chambers that are more inflated, making the trajectory of the center of mass moves with a curvature radius determined by this offset and the friction with the ground. Finally, the amplitude A limits the maximum pressure during the oscillation, thereby controlling the snake's speed. The parameters that make the snake move forward depend on the physical properties of the snake, such as weight, length, friction coefficient with the floor, and were determined experimentally in [11]. This controller is an open-loop method, which generates the forward motion and allows to make turns, but no feedback is given if the trajectory is deviating from the desired trajectory.

The pressure delivery is not instantaneous but limited by the maximum airflow allowed by the valves. Assuming the pressure source can reliably maintain a constant output pressure p_s , the air flow v to the chamber is given by [30],

$$v^2 = \frac{2}{\rho}(p_t - p_s), \quad (10)$$

where ρ is the air density and p_t is the pressure in the chamber. This means the pressure update is proportional to the square of the difference between the current pressure and the desired pressure. The pressure update for inflation in each step is then defined based on the difference Δp_i , in Eq. 11.

$$\Delta p_i = \frac{a_i(t+h) - p_i(t)}{p_s}, \quad (11)$$

The deflation releases pressure in the atmosphere while keeping its own pressure relatively constant due to the change of volume. Therefore, the deflation ratio should be close to linear up to a threshold T_p when it is proportional to the over-pressure with a damping,

$$p_i(t+h) = \begin{cases} p_i(t) + p_s \Delta p_i^2 k_i & \text{is inflating} \\ p_i(t) - \min(p_i(t)k_d, T_p) & \text{is deflating} \end{cases} \quad (12)$$

where $k_i, k_d \in (0, 1]$ are the inflation damping parameters, and are tuned according to the experimental data.

VI. EXPERIMENTAL VERIFICATION

For all tracking experiments, the poses of the rigid links were captured using the motion capture system (MOCAP) by placing four markers on each rigid extremity so that it can collect their full poses. The MOCAP system contains 11 cameras surrounding the observable space of $4 \times 3 m^2$. This redundancy allows the information of links to be collected with high precision and minimal loss of tracking during the experiments. In order to eliminate remaining outliers, every experiment is repeated for 10 times unless otherwise mentioned. To ensure data sanity, every sample that deviates more than an expected maximum displacement from the collected values is pruned and not used in the analysis, as this type of samples results in a data collection with negligible standard deviation. Fig. 1 shows the markers on the top corners of the rigid plates.

A friction coefficient of $\mu_i = 1$ has been used for all experiments. We implement our simulator in CUDA and run it on a computer with an Intel Core-i7 8520K, 32GB of RAM, and one NVIDIA GTX 1080 Ti GPU. We use a fixed time-step of $h = 0.0083s$, each time-step performs 4 Newton iterations, with each linear system solved approximately using 20 PCR iterations to ensure a fixed computational cost.

A. Quasi-static Verification

The first experiment on the simulation is to verify whether the pressure actuator follows the same geometrical behavior as the real link. For this experiment, the curvatures of the real links were obtained by subtracting the yaw of the rigid connectors attached to each soft link, for the varying overpressures (the pressure that exceeds the resting atmospheric

pressure) from 0 to 10 psi, moving up with steps of 1 psi for both directions on the link, and averaged over 300 samples. Negative values show the inflation of left chamber and positive values are for the right chamber.

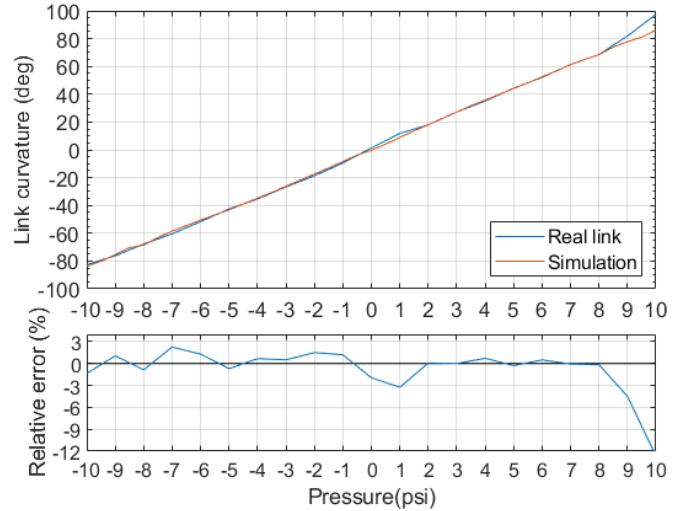


Fig. 5. Curvature vs. Pressure for simulation and a real link. The top plot shows that the curvature follows a linear relationship with the pressure applied. The bottom plot is the relative error.

From Fig. 5, it can be seen that the curvature increases linearly with the pressure within a range. Particularly, the spring model is able to closely match the real curvature, and accurately follows the linear model up to 8psi. Ecoflex 00-30 Young modulus is 66 kPa, at 9 psi (62 kPa) it nears 2x expansion, which is the limit at which the material is linear. This behavior is also clearly observed by the relative error plot. When the pressure exceeds 8 psi, the real link starts to bulge over imperfections in the manufacturing, and on the opposite side, it folds in itself, resulting in a deviation from the linear model. Besides, since the extra pressure is forcing over the imperfections of the manufacturing process, there is a potential risk of damaging the links in the long term. For these reasons, it was deemed that the safe pressure threshold shall be 8 psi, and all the remaining tests were restricted to up to that range.

B. Dynamic Verification

The dynamic verification starts with the analysis to the step response on the actuators. A single link was used to capture the rate at which it inflates and deflates from rest to 60 – 100% at 10% steps. These trials were used to tune the gains $k_i = k_d = 0.23$ and $T_p = 0.68$. The results are seen in Fig. 6.

The simulator trajectory is then tested with an open-loop generator from Eq. 9 and compared with the real snake robot using the following parameters $\omega = 2Hz$, $\alpha = \pi/2$, $\psi = 0$ and $A = p_{max} = 8psi$. Since it's an open-loop control, it is expected that due to model inaccuracies and unmodeled dynamics the simulator will have error accumulated along the trajectory. As a result, the simulated trajectory may diverge from the actual trajectory over time. The comparison



Fig. 4. Visual comparison of link expansion after settling from -10 to 10 psi overpressure. Negative values mean the left chamber is inflated, while positive values are for the right chamber. The simulation displays high accuracy on the curvature up to 8 psi, where the dashed lines were traced. After that the pressure becomes excessive and the real link stops following the linear model (best seen in digital format).

Snakes (#)	1/4	1	2	3	4	5	6	7	8	9	10
Assembly (ms)	0.74	1.11	1.41	2.33	3.41	5.59	6.19	6.45	7.37	8.48	10.31
Solve (ms)	4.96	10.52	25.06	31.90	43.12	50.41	58.63	66.95	79.73	86.89	95.57
Total (ms)	5.70	11.63	26.47	34.23	46.53	56.00	64.82	73.4	87.10	95.37	105.88
Total/Snake (ms)	-	11.63	13.23	11.41	11.63	11.2	10.80	10.48	10.88	10.59	10.58

TABLE II
BENCHMARK RESULTS FOR OUR SIMULATOR

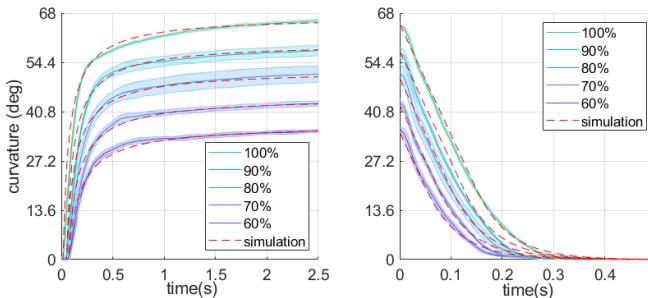


Fig. 6. Step analysis for chamber inflation (left) and deflation (right) with approximate response from simulator.

was made using the center of mass of the snake. The divergence can be addressed with the use of closed-loop feedback control for both simulated snake and the real snake robots. However, from Fig. 7, it can be seen that starting from the same initial condition, the simulated trajectory closely matches the real trajectory for the execution time. The inclusion of the latency model for the pressure update makes the trajectory of the center of mass go in the same overall direction and amplitude as the real snake.

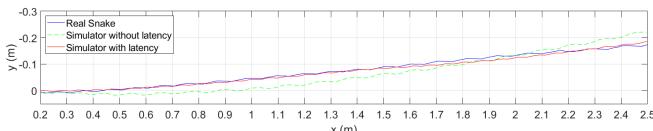


Fig. 7. Trajectory comparison between collected data (blue), simulator without actuator latency (green), and with latency (red). The latency model in the pressure update makes the simulator's trajectory be closer to the real snake under same conditions (best seen in digital format).

C. Benchmarking

In order to test scalability of the system, we benchmark simulation times for a single soft link, a full snake with 4 links, and up to 10 snakes, each with 4 links. The linear system for a single Newton iteration corresponding to a single snake is a sparse matrix of roughly 30000×30000 .

Each frame is sub-divided in two sub-steps for the constraint solver, and the optimization runs with 20 iterations each. The total per-frame simulation times are in Table VI. From the results, it can be seen that simulation time increases linearly with the number of links and snakes. Linear scaling is expected, since a single link saturates the GPU. On average, it takes less than 12 ms to simulate each snake, with optimal performance obtained when simulating seven snakes together. One single snake takes 8.9s per frame running at 3.3GHz on an Intel Core i7 5820k.

VII. CONCLUSION

We have presented a dynamical model for simulating 1D pneumatic actuators and a framework to simulate it in a multi-physics environment in real time. By comparing the simulation with a real soft robotic snake, it is demonstrated that the simulated snake produces real-time high-fidelity results even in complex scenarios involving a mixture of hybrid soft bodies, rigid bodies, and friction contacts. In the open loop control analysis, it becomes clear that the simulator doesn't perfectly model the real dynamics; however, it remains in close performance to the real snake. Our next step is now apply a control trained and tested in simulation on the real snake. The transfer from simulation to reality, while still maintaining stability, can also be facilitated by domain randomization techniques [20]. We demonstrated the use of GPU in accelerating high-fidelity simulation for soft robots and present a framework that can generalize to other soft robotic systems. Our future research is to develop learning-based control to train fast and stable snake gaits in a range of terrains, as well as obstacle-aided navigation, and learning specific motion primitives from demonstration [31, 32]. Another step is also study the model different types of soft actuators such as PneuNet [2], and plan to publish data for our snake model to enable an open platform for experimentation and improvement.

REFERENCES

- [1] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, “Soft robotics for chemists,” *Angewandte Chemie*, vol. 123, no. 8, pp. 1930–1935, 2011.
- [2] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, “Soft robotics: Biological inspiration, state of the art, and future research,” *Applied bionics and biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.
- [3] M. Luo, M. Agheli, and C. D. Onal, “Theoretical modeling and experimental analysis of a pressure-operated soft robotic snake,” *Soft Robotics*, vol. 1, no. 2, pp. 136–146, 2014.
- [4] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt *et al.*, “Software toolkit for modeling, simulation, and control of soft robots,” *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.
- [5] A. Rodriguez, E. Coevoet, and C. Duriez, “Real-time simulation of hydraulic components for interactive control of soft robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4953–4958.
- [6] O. Goury and C. Duriez, “Fast, generic and reliable control and simulation of soft robots using model order reduction,” *IEEE Transactions on Robotics*, pp. 1–12, 2018.
- [7] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *International Conference on Machine Learning*, 2015, pp. 1889–1897.
- [8] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation,” *CoRR*, vol. abs/1506.02438, 2015.
- [9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *CoRR*, vol. abs/1707.06347, 2017.
- [10] M. Neunert, T. Boaventura, and J. Buchli, “Why off-the-shelf physics simulators fail in evaluating feedback controller performance—a case study for quadrupedal robots,” in *Advances in Cooperative Robotics*. World Scientific, 2017, pp. 464–472.
- [11] S. Ozel, E. H. Skorina, M. Luo, W. Tao, F. Chen, Y. Pan, and C. D. Onal, “A composite soft bending actuation module with integrated curvature sensing,” in *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2016, pp. 4963–4968.
- [12] M. Luo, E. H. Skorina, W. Tao, F. Chen, S. Ozel, Y. Sun, and C. D. Onal, “Toward modular soft robotics: Proprioceptive curvature sensing and sliding-mode control of soft bidirectional bending modules,” *Soft robotics*, vol. 4, no. 2, pp. 117–125, 2017.
- [13] M. Luo, Y. Pan, W. Tao, F. Chen, E. H. Skorina, and C. D. Onal, “Refined theoretical modeling of a new-generation pressure-operated soft snake,” in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2015, p. V05CT08A023.
- [14] C. Nvidia, “Cusparse library,” *NVIDIA Corporation, Santa Clara, California*, 2014.
- [15] M. Manti, T. Hassan, G. Passetti, N. D’Elia, C. Laschi, and M. Cianchetti, “A bioinspired soft robotic gripper for adaptable and effective grasping,” *Soft Robotics*, vol. 2, no. 3, pp. 107–116, 2015.
- [16] I. D. Walker, “Continuous backbone ‘continuum’ robot manipulators,” *ISRN Robotics*, 2013.
- [17] P. Polygerinos, Z. Wang, J. T. B. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, “Modeling of soft fiber-reinforced bending actuators,” *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 778–789, June 2015.
- [18] C. Duriez, “Control of elastic soft robots based on real-time finite element method,” in *IEEE International Conference on Robotics and Automation*, May 2013, pp. 3982–3987.
- [19] M. Pozzi, E. Miguel, R. Deimel, M. Malvezzi, B. Bickel, O. Brock, and D. Prattichizzo, “Efficient fem-based simulation of soft robots modeled as kinematic chains,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–8.
- [20] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, “Sim-to-real: Learning agile locomotion for quadruped robots,” *CoRR*, vol. abs/1804.10332, 2018.
- [21] M. Servin, C. Lacoursiere, and N. Melin, “Interactive simulation of elastic deformable materials,” in *Proceedings of SIGRAD Conference*, 2006, pp. 22–32.
- [22] D. E. Stewart and J. C. Trinkle, “An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.
- [23] A. Shabana, *Computational Dynamics, 3rd Edition*. Wiley, 2009.
- [24] D. E. Stewart, “Rigid-body dynamics with friction and impact,” *SIAM review*, vol. 42, no. 1, pp. 3–39, 2000.
- [25] E. Todorov, “Implicit nonlinear complementarity: A new approach to contact dynamics,” in *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2010, pp. 2322–2329.
- [26] T. S. Munson, F. Facchinei, M. C. Ferris, A. Fischer, and C. Kanzow, “The semismooth algorithm for large scale complementarity problems,” *INFORMS Journal on Computing*, vol. 13, no. 4, pp. 294–311, 2001.
- [27] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.
- [28] J. Di, S. Yao, Y. Ye, Z. Cui, J. Yu, T. K. Ghosh, Y. Zhu, and Z. Gu, “Stretch-triggered drug delivery from wearable elastomer films containing therapeutic depots,” *ACS nano*, vol. 9, no. 9, pp. 9407–9415, 2015.

- [29] H. Si, “Tetgen: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator,” *Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany*, 2006.
- [30] F. White, *Fluid Mechanics*, ser. McGraw-Hill series in mechanical engineering. McGraw-Hill, 2008.
- [31] A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots: a review,” *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [32] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.