Contact-implicit trajectory planning for continuum manipulators

Moritz A. Graule and Clark B. Teeple

May 4, 2018

Abstract

In this work, we tackled kinematic planning of soft robots using contact along the whole manipulator. We extended the Smooth Curvature Model for flexures to include contact points along the length. As a first step toward kinematic trajectory planning, we then formulated a shape planning problem with one segment of a continuum manipulator that needs to reach a target end-effector pose that is behind an obstacle. Formulating this problem as a constrained optimization, we are able to achieve reasonably close end-effector poses that rely on contact with the obstacle to get closer to the target pose. Finally, in preparation to evaluate the resulting kinematic plans, we designed, built, and characterized a soft robotic manipulator arm, which exhibits kinematics that follow our model.

1 Introduction

Soft continuum robots are well suited for situations where robots need to interact gently with objects in their environments. Compliance in the structures of soft robots enables passive adaptation to uncertainties in positioning, object shape, and environmental constraints, making them ideal for use around humans and in delicate environments [1, 2]. In addition, complicated motions and contact interactions can be generated with soft robots using minimal high-level control effort compared to rigid robots. This "mechanical intelligence" can provide enhanced dexterity with minimal computational cost for actions such as hyper-redundant manipulation and grasping [3, 4, 5, 6, 7].

While teleopration of soft robots can enable robust manipulation, they become more useful in manufacturing and home use if they can operate autonomously. To enable a soft robot to reason about its kinematics to reach some goal state, high-level trajectory planning is key. However, trajectory planning for soft robots is challenging because continuous deformation in soft bodies can lead to a huge number of degrees of freedom which are hard to capture in a low-dimensional model. In addition, while soft robots are well-suited for contact with the world, the typical family of models used to describe their shape are insufficient to robustly handle contact along the length, as shown in Fig. 1.

In this work, we break the problem into three parts. We first develop a backbone model for bending of a soft element that incorporates contact forces directly, based on the work of Odhner and Dollar [8]. Next, to exemplify the utility of our model for trajectory planning, we formulate a kinematic planning problem for a soft robot with an obstacle. Finally, in preparation to evaluate the resulting kinematic plans, we develop and characterize a soft robotic manipulator arm, and verify its kinematics follow our model.

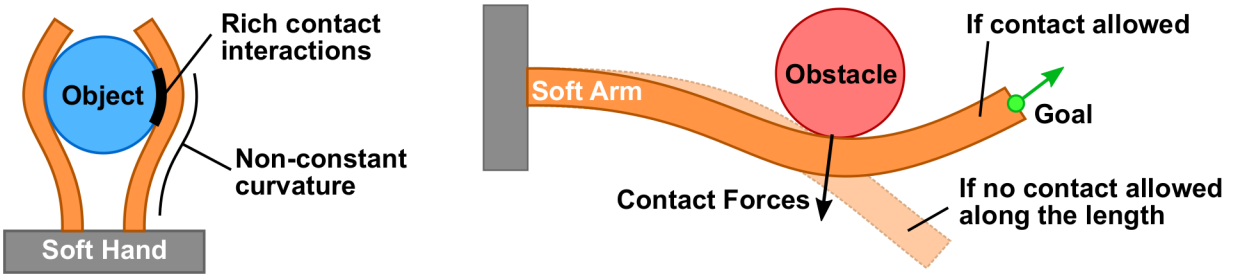


Figure 1: Soft robots are capable of handling rich contact interactions with objects and the environment with ease. However, in the presence of contact, constant curvature assumptions are insufficient to describe the higher-order bending that occurs.

2 Background and Related Work

2.1 Modeling the kinematics of continuum robots

Modeling of soft continuum robots traditionally involves considering a central bending axis (backbone) and an envelope around this backbone describing the physical dimensions of the system. In this section, we will discuss the most widely used continuum actuator model along with the strengths and weaknesses of this approach.

By far the most common model for the backbone shape of a continuum actuator is piecewise constant curvature (PCC) [9]. In PCC, we assume each actuator segment has a constant radius of curvature, and that actuators can then be stacked piecewise using coordinate transformations from the base of each segment to the tip. By assuming each segment has constant curvature, we restrict the state space of a planar robot to include only one state per actuator segment (assuming constant backbone length), the curvature of the segment. The coordinate transformation from base to tip is then fully defined, and a straightforward extension allows the backbone to change in length.

2.2 Trajectory optimization for soft robots

Due to its low dimensionality, PCC has been applied in trajectory planning as a convenient kinematic representation for soft robots. For example, PCC with a fixed-length backbone was applied to a horizontal planar arm with minimal error in end-effector tracking, however the arm was not allowed to come in contact with the environment. [10]. An extension of this work used PCC with an envelope to do full-arm trajectory planning for operation in confined spaces [11]. In this application, contact with the environment was not modeled explicitly, but instead the compliance of the arm was utilized to "harmlessly collide" with the environment in the presence of position errors. Trajectory planning for soft robots was also extended to three-dimensions with similar results [12].

PCC was also used to perform trajectory optimization, but still not considering contact. By representing the dynamics of soft actuators using lumped masses at the center of each actuator and the kinematics using PCC, trajectory optimization of a soft manipulator was performed in the vertical plane [13]. In addition bracing against surfaces in the environment was shown to enable a soft manipulator to reach regions of the state space previously unattainable. In this case, contact forces were incorporated into the model, but PCC was still used to represent the kinematics even when the robot was in contact because backbone deformation was relatively small.

More recently, impedance control was demonstrated using a soft robot to enable the end-

effector to move while in contact with a surface [14]. However, forces were applied only at the end effector, so again PCC was used to describe the kinematics of the robot with minimal error.

3 Project Overview

The problem we want to solve in this project is kinematic planning of soft robots using contact along the length. One large challenge in formulating trajectory planning for soft robots is modeling the kinematics and dynamics of highly-deformable structures in a low-dimensional space. While piecewise constant curvature models work for soft robots that have only small contact interactions, PCC quickly becomes inaccurate when any appreciable external forces are applied along the length of the robot, as shown in Fig 1. These inaccuracies render PCC insufficient for planning of soft robot trajectories when they interact with their environments.

Thus, we need a way to formulate a low-dimensional kinematic representation for soft robots that directly incorporates contact points along the length. We can turn to the work of Odhner et. al [8] on the Smooth Curvature Model for flexures. Odhner et. al. model flexible members in a mechanism using third-order Legendre polynomials for curvature of a beam, then consider an energy balance in the system using the Lagrangian. In addition, they include an external force and moment at the tip of the flexure. We extended the Smooth Curvature Model to include contact forces acting normal to the backbone, as discussed in section 4.

Next, to test the feasibility of our model for kinematic planning, we designed a planning problem where a soft robot needs to achieve an end-effector position and orientation in the presence of an obstacle. The robot can use contact with the obstacle to achieve a goal state. We formulated this as an optimization problem as discussed in section 5.1.

Finally, we want to test the relevance and feasibility of our model with real soft robots. To do this, we designed and built a planar soft robotic manipulator arm made of modular bending/extension elements, and characterized the kinematics of each segment under pressure with no load. While we have not finished verifying our model and trajectory planning on this platform, the design of this platform is discussed in section 5.2.

4 Contact-implicit Smooth Curvature Model

We approximate the backbone of a segment using a third order Legendre polynomial (as used for the representation of flexures in [8]) as

$$\hat{\omega}(q,s) = \frac{\alpha_1}{L} + \frac{\alpha_2}{L}\left(\frac{2s}{L} - 1\right) + \frac{\alpha_3}{L}\left(\frac{6s^2}{L^2} - \frac{6s}{L} + 1\right)$$

This allows us to represent the system in the generalized coordinates

$$q = [\alpha_1, \alpha_2, \alpha_3, L]$$

where L is the length of the backbone and α_i are the coefficients of the Legendre polynomials. The angle of the backbone can be expressed as a function of the position along the curve s (with $0 \leq s \leq q(4) = L$) as

$$\phi(q,s) = \alpha_1 \frac{s}{L} + \alpha_2\left(\frac{s^2}{L^2} - \frac{s}{L}\right) + \alpha_3\left(\frac{2s^3}{L^3} - \frac{3s^2}{L^2} + \frac{s}{L}\right)$$

. The global coordinates of a point on the beam at $s = s_c = cL$ with $0 \leq c \leq 1$ are then

$$x_c(q) = \int_0^{s_c} \cos(\phi(q,s))ds = \int_0^{cL} \cos(\phi(q,s))ds$$
$$y_c(q) = \int_0^{s_c} \sin(\phi(q,s))ds = \int_0^{cL} \sin(\phi(q,s))ds$$

and the tip angle is

$$\phi_t(q) = \phi_t(q,L) = \alpha_1$$

The potential energy V stored in this system consists of energy from backbone curvature V_c, backbone extension V_e, fluid pressure, and gravitation; its kinematic energy T consists of the acceleration of the module's center of mass and fluid flow. Assuming operation at steady-state conditions within a horizontal plane allows us to express the the Lagrangian of the system as

$$\mathcal{L} = T - V = -V_c - V_s$$

with

$$V_c = \frac{EI}{2L}\left(\alpha_1^2 + \frac{\alpha_2^2}{3} + \frac{\alpha_3^2}{5}\right)$$

and

$$V_e = \frac{EA_{eff}}{2}(L_0 - L)^2$$

where I is the moment of inertia of the module, E the effective Young's modulus, and A_{eff} the effective area.

Given a number of i contact points $P_{c,i} = [x(s_{c,i}), y(s_{c,i})]^T = [x_{c,i}, y_{c,i}]^T$ along the module, with contact forces $\mathbf{F}_{c,i} = F_{c,i}[\sin(\phi_c), -\cos(\phi_c)]^T$, and a tip moment M_a (from the actuators), the following equation has to be fulfilled in equilibrium:

$$\frac{\partial \mathcal{L}}{\partial q} + \sum_i J_{ci}^T \mathbf{F}_{ci} + \frac{\partial \phi_t}{\partial q} M_a = \mathbf{0}$$

Here, J_{ci} is the Jacobian that transfers the contact forces into generalized forces (see e.g. [8] for more details); ϕ_t is the tip angle. The Jacobian can be computed as

$$J_c = \frac{\partial P_c}{\partial q} = \begin{bmatrix} \frac{\partial x_c}{\partial \alpha_1} & \frac{\partial x_c}{\partial \alpha_2} & \frac{\partial x_c}{\partial \alpha_3} & \frac{\partial x_c}{\partial L} \\ \frac{\partial y_c}{\partial \alpha_1} & \frac{\partial y_c}{\partial \alpha_2} & \frac{\partial y_c}{\partial \alpha_3} & \frac{\partial y_c}{\partial L} \end{bmatrix}$$

The actuator moment can be computed from the two actuator forces $N_{a1,a2}$ as $M_a = d(N_{a2} - N_{a1})$, where d is the actuator moment arm, a design parameter. Details on computing the relevant derivatives are provided in Appendix C. Without loss of generality, contact forces are here defined to be normal to the backbone.

5 Toward Trajectory Planning Using Contact

To enable later work on full trajectory planning, we first formulated final state planning given a desired objective, and we also designed, built, and characterized a physical prototype platform. These two components are critical to ensure our model works for shape planning, while simultaneously preparing for testing of shapes and trajectories on a real system.

5.1 Shape Planning Problem Specifications

We addressed the two problems of planning for desired manipulator configuration using actuator torques and forces, and planning for a desired end effector position in an environment with an obstacle. We formulated these two tasks as constrained optimizations. Details on the task of planning without obstacles are omitted from this report, as that problem is a simpler version of the one discussed below.

At the heart of both optimizations is a constraint that enforces a feasible equilibrium based on equation 1 as

$$g_e(q) = \frac{\partial \mathcal{L}(q)}{\partial q} + \sum_i J_{ci}^T \mathbf{F}_{ci} + \frac{\partial \phi_t}{\partial q} M_a \tag{1}$$

We empirically determined that it is beneficial to remove the fourth row of $g_e(q)$ and instead add an additional constraint which directly enforces the correct backbone length as

$$L = L_0 + \frac{N_a}{A_{eff} E}$$

In this section, we plan for a manipulator configuration that brings the end effector as close as possible to a desired pose (position and orientation, designated as $\mathbf{P}_{t,d}$). We introduce an obstacle into our planning environment, with which the manipulator is not allowed to overlap. The obstacle is defined by its center coordinates $[x_o, y_o]$ and its radius r_o. We optimize over the actuator forces (N_{a1}, N_{a2}), one contact force with the magnitude (F_{c1}) and position along the curve (c_1), and manipulator configuration (q). The complete optimization is shown below, followed by an explanation of the cost terms and constraints.

$$\begin{aligned}
\min_{[N_{a1}, N_{a2}, F_{c1}, q, y]} \quad & u^T R u + (\mathbf{P}_{t,d} - \mathbf{P}_t)^T Q (\mathbf{P}_{t,d} - \mathbf{P}_t) + (\mathbf{P}_o - \mathbf{P}_c)^T Q_c (\mathbf{P}_o - \mathbf{P}_c) + \beta_1 y - \beta_2 F_c \\
\text{s.t.} \quad & L - L_0 - \tfrac{N_a}{A_{eff} E} = 0 \\
& g_e(q) = 0 \\
& -c_1 \leq 0 \\
& c_1 - 1 \leq 0 \\
& \Phi(q, s_{check}) \leq 0 \; \forall s_{check} = [0, 1, ..., N_{check}] \tfrac{q(4)}{N_{check}} \\
& \Phi(q, s_c) = \Phi(q, cq(4)) \leq 0 \\
& -\mathbf{F}_c \cdot (\mathbf{P}_c - \mathbf{P}_o) \leq 0 \\
& \Phi(q, s_c) F_c - y = 0 \\
& -y \leq 0
\end{aligned} \tag{2}$$

The cost penalizes the magnitude of actuator forces ($u^T R u$) and the deviation from the desired tip pose ($(\mathbf{P}_{t,d} - \mathbf{P}_t)^T Q (\mathbf{P}_{t,d} - \mathbf{P}_t)$). A term rewarding proximity between the contact point on the backbone and the obstacle ($(\mathbf{P}_o - \mathbf{P}_c)^T Q_c (\mathbf{P}_o - \mathbf{P}_c)$) and a term rewarding increasing contact forces ($\beta_2 F_c$) were added to the cost in order to drive the convergence towards solutions utilizing contact. The cost further includes a term with a slack variable ($\beta_1 y$) to relax the contact complimentary constraint ($\Phi(q, s_c) F_c$).

The constraints contain the equilibrium constrains as outlined above; constraints ensuring that the contact point is on the backbone ($0 \leq c_1 \leq 1$); a guard function ($\Phi(q, s)$) ensuring that the manipulator cannot penetrate the obstacle; a constraint enforcing that the contact force $\mathbf{F}_c$ points away from the obstacle ($-\mathbf{F}_c \cdot (\mathbf{P}_c - \mathbf{P}_o) \leq 0$); and a complementarity constraint ($\Phi(q, s_c) F_c -$

$y = 0$) which ensures that contact forces can only be non-zero when the contact point lies on the obstacle and the manipulator. The guard function

$$\Phi(q, s_c) = -|\mathbf{P}_s - \mathbf{P}_o| + r_o$$

with obstacle position $\mathbf{P}_o = [x_o, y_o]^T$ and point on manipulator at s $\mathbf{P}_s = (x_s, y_s)$ is evaluated at N_{check} equally spaced check points along the backbone to ensure that no point of the manipulator lies inside the obstacle region ($N_{check} = 20$ provided sufficient performance). A slack variable y was introduced in the complementarity constrain to improve the convergence behavior of the optimization; this variable is driven to zero by the respective term in the cost ($\beta_1 y$).

5.2 Model verification using hardware

Two-dimensional soft actuator pairs (as shown in Fig. 2) will provide a simple physical platform to evaluate our model for a generic soft robot. An actuator pair (hereby called a bending module) can bend in both directions by pressurizing either actuator, and extend by pressurizing both actuators together. Rigid couplings allow stacking of an arbitrary number of modules in serial to create a planar manipulator.

To model the behavior of each actuator, we will treat the extension actuators as linear pistons with a spring-damper system attached, as shown in Fig. 2b. We consider the internal pressure $P(t)$, the piston area $A(t) = A$, the volume of air in the piston $V(t)$, the height of the piston $h(t)$, and the spring-damper system parameters k and c. At a pressure of 0, the spring is at resting length. We can write the force balance on the top plate to be:

$$P(t)A(t) - kz - c\dot{z} = 0 \tag{3}$$

where $z(t) = h(t) - h_0$ is the extension and h_0 is the initial height. Thus, we see that we have a second-order linear system.

Now we can formulate a model for a bending module that fits into our framework for generic continuum actuators. We can model the tip location of each bending module with respect to the root can be represented as a translation in x and z, and a rotation θ. This is a useful coordinate definition to enable chaining of modules through a coordinate shift. Then we can approximate the forces applied to the tip plate via the two actuators with a resultant force N_a acting perpendicular to the plate, and a moment M_a. Finally, we can define contact forces F_{ci} acting at a points s_{ci} along the backbone. We assume for now that the contact forces act normal to the backbone.

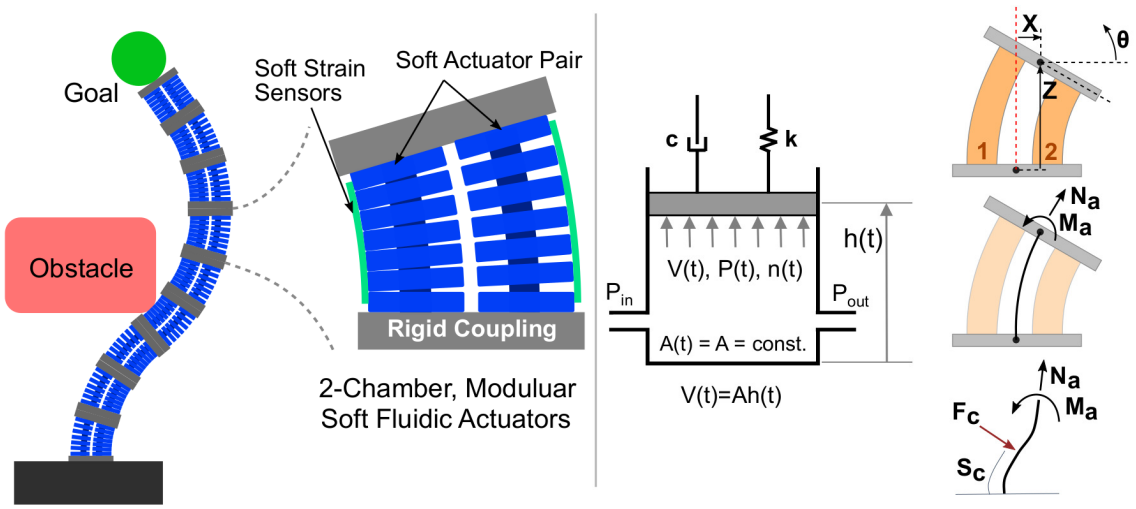
6 Experiments

6.1 Simulation

We solved the optimizations outlined in section 5.1 using MATLAB's (MathWorks, Natick MA) fmincon function. We tested various configurations of the objective function (initially omitting many of the terms present in its final form) and many different cost parameters to identify a optimization setup that converges frequently.

6.2 Physical Robot

Several calibration steps were necessary to obtain an accurate model of actuator and bending module behavior. We measured the effects of input pressure on free extension and blocked force for this actuator design, and characterized the viscoelastic behavior of the actuators using a force relaxation test. We then characterized the in-plane bending of a two-actuator module.



(a) (b)

Figure 2: Proposed two-dimensional soft robotic manipulator platform in the presence of obstacles, with associated actuator models. The soft arm (a) is made up of two-chamber actuator modules stacked in a serial configuration, allowing end-effector motion in a plane. Each actuator will be modeled as a piston with spring-damper attached (b). The tip location of each bending module can be represented as a translation in x and z, and a rotation θ. The bending modules can then be represented by a backbone curve with resultant actuator force (N_a) and moment (M_a) at the tip, and contact forces F_{ci} applied along the length.

6.2.1 Characterization of single actuators

To characterize the static and dynamic response of individual extension actuators, we performed blocked force and free extension tests along with force relaxation tests.

Blocked force was measured for one actuator using the load cell of an Instron uniaxial testing machine. First, we attached each actuator to a custom-designed 3D-printed (Markforged Mark Two) fixture clamped in the jaws of the Instron. Next, air pressure was applied between 0 and 40 psi using a manual regulator and digital pressure sensor. Actuator pressure and the resulting force on the load cell were then measured simultaneously using the Instron for three separate tests.

Free extension as a function of input pressure for six actuators was measured using digital calipers. We applied air pressure to each actuator using the same procedure detailed for blocked force, ensuring the actuators were settled before taking measurements. Afterward we found the extension by subtracting the length of the actuators at each pressure by the original length at 0 psi.

To measure the dynamics of the actuators, we performed force relaxation tests. To begin, an actuator was mounted to the Instron using the same fixtures as in the blocked force tests. Using the Instron crosshead, a step extension was applied (30 mm) and the resulting force over the next 30 seconds was measured. Using this time response, we can evaluate the damping behavior of the TPU material.

6.2.2 Kinematics of two-actuator bending modules

After testing single actuators, we tested the kinematics of a two-actuator module with no loads applied by pressurizing both actuators to various pressures and measured the resulting tip state (position and orientation). For a single module, we applied input pressures from 0 to 40 psi into both actuators, and took pictures of the module from an overhead view to recover its shape. A

sample of pictures taken is shown in Fig. 3. Next, we used ImageJ (U. S. National Institutes of Health, Bethesda, Maryland, USA) to measure the positions of the tip with respect to the base for each image. The results are shown and discussed below.

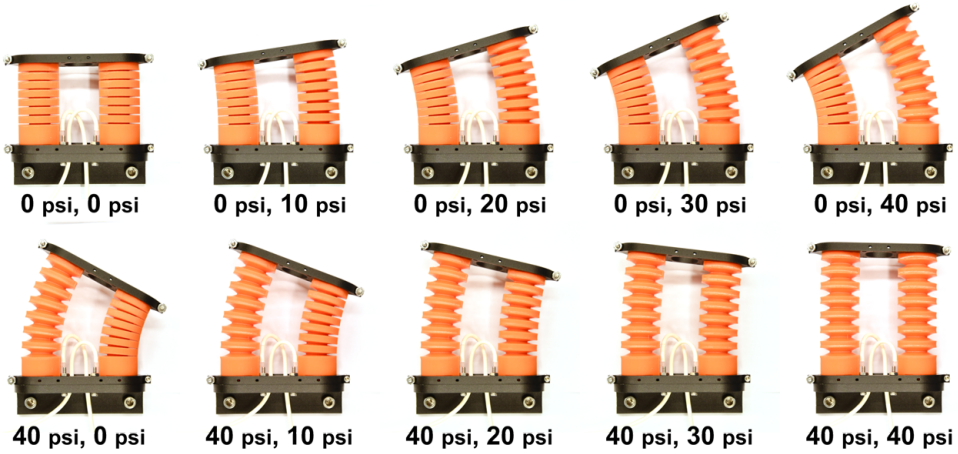


Figure 3: Planar modules bend and extend as a function of input pressure to the two actuators.

7 Results

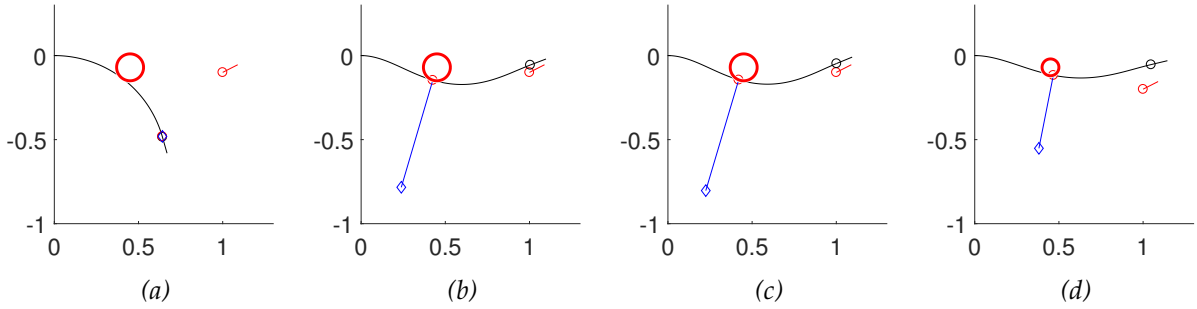
7.1 Simulations

The final shape of a soft bending segment relies on contact force with the environment to reach the desired end-effector pose is shown in Fig. 4.

7.2 Physical Robot

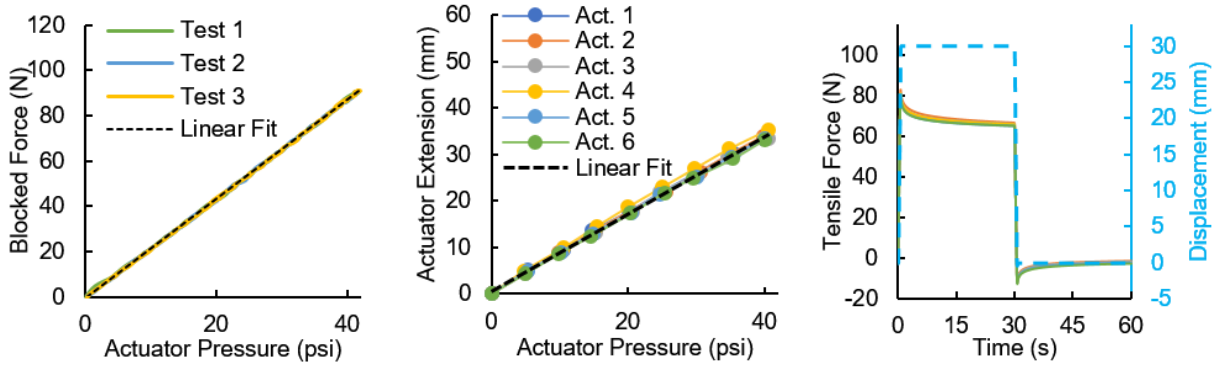
Static and dynamic characterization of single extension actuators verify our simple piston model. As shown in Fig. 5, linear relationships were found from input pressure to both blocked force and free extension. This indicates that our simple actuator model (piston w/ spring-damper) is sufficient to describe forces and displacements of actuators. In addition, force relaxation as a function of a step in displacement shows a typical exponential decay, indicating that the actuators have approximately viscous damping, thus fully validating our actuator model.

Next, static characterization of a bending module shows that we can make a few simplifying assumptions about the module. First, linear relationships are shown for the tip angle and lateral deviation as a function of the pressure differential (difference between actuator pressures), as shown in Fig. 6. Next, a roughly linear relationship exists between the extension and the total input pressure (addition of the two actuator pressures). These three relationships allow us to assume the module has two single-actuators that simply apply a force and a moment on the top plate. This validates our model for a single module as specified above in Fig. 2b.



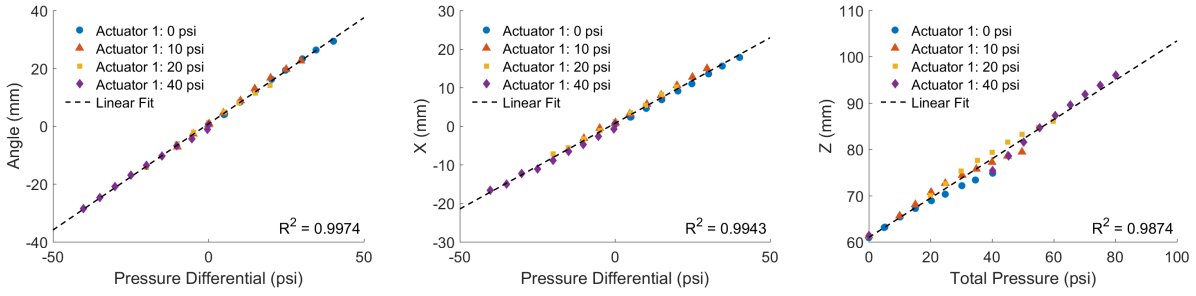
(a) *(b)* *(c)* *(d)*

Figure 4: The final shape of a soft bending segment (black) relies on contact force (blue) with the environment (red circle) to reach the desired end-effector pose (red). When contact force is constrained to be zero, but objects collisions are still in place (a), the resulting shape cannot reach the goal. However, when complementarity is implemented and the contact force at the object is allowed to be nonzero (b-d), reasonably close solutions are found.



(a) *(b)* *(c)*

Figure 5: Characterization of single extension actuators yields linear relationships for both blocked force (a) and extension (b) as a function of input air pressure. In addition, force relaxation (c) as a function of a step in displacement shows a typical exponential decay, indicating that the actuators have viscous damping



(a) *(b)* *(c)*

Figure 6: Characterization of bending modules under no load and not contact yields linear relationships for all quantities. Linear relationships exist for the tip angle (θ) and lateral displacement (x) as a function of pressure differential between the two actuators ($P_1 - P_2$), and the extension (z) as a function of total pressure ($P_1 + P_2$).

8 Discussion

8.1 Simulations

The current implementation requires good initial guesses and sometimes tuning of the parameters in the cost function. To obtain a desirable solution, it is generally necessary to run the optimization a few times, as there are many local optimums which do not utilize the contact forces.

8.2 Physical Robot

Since our soft robot platform can be modeled using the representation specified, this means it is valid to plan its motion using our backbone model for a generic continuum manipulator. Thus, we have created a platform that can be used to verify our high-level model and its efficacy in trajectory planning.

9 Future Work

We will refine our optimization setup further to increase the convergence rate; expand the setup towards kinematic planning (i.e., finding intermittent equilibrium configuration spaced between an initial and a target configuration); and combine multiple segments (it is straightforward to obtain the transformations between segments from the tip position and orientation). We will further transfer our optimization task to a different solver.

Future work on our soft arm platform will include implementation of shape and trajectory planning based on results of our simulations. We plan to first use a Vicon motion capture arena to collect the shape of the actuators over time. In addition, we can use the Vicon to perform closed-loop trajectory following. Finally, we plan to execute closed-loop control using on-board sensing of each actuator's extension state.

Acknowledgments

Thank you to Daniel Vogt at Harvard's Wyss Institute for insightful discussions and help with the initial actuator and platform design.

A System Description (CODE)

In order to run our optimization, please put all submitted MATLAB files into one folder and run the file *run_optimization_v6_submission.m*. Desired end effector positions and obstacle definitions can be entered in the beginning of this file. Rerun if the optimization does not converge to a desirable solution.

All functions and derivatives used in our code are provided in analytical form in C.

B Group Makeup

- Moritz Graule
  - Developed the kinematic model

  - Developed the kinematic planning problems and simulations
  - Helped design the physical robotic platform
  - Helped characterize actuators and actuator module behavior
- Clark Teeple
  - Helped develop the kinematic model
  - Helped develop and run simulations
  - Designed and fabricated the physical robotic platform
  - Characterized actuators and actuator module behavior

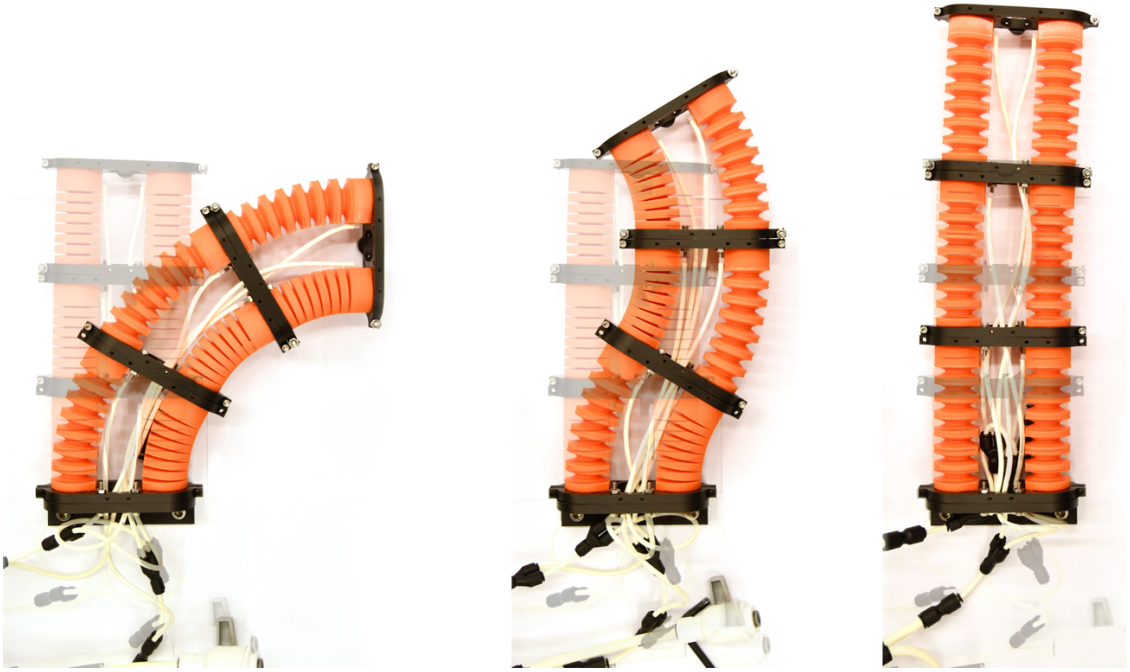


Figure 7: Three-module manipulator can display unidirectional bending (left), complex bending (center), and extension (right). Pressurized actuators were set to 40 psi.

References

[1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[2] J. Hughes, U. Culha, F. Giardina, F. Günther, and A. Rosendo, "Soft Manipulators and Grippers: A Review," *Frontiers in Robotics and AI*, vol. 3, no. November, pp. 1–12, 2016.

[3] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[4] ——, "Soft hands for reliable grasping strategies," *Soft Robotics: Transferring Theory to Application*, pp. 211–221, 2015.

[5] 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.

[6] R. C.-h. Yeow, J. H. Low, W. W. Lee, P. M. Khin, S. L. Kukreja, S. Member, and H. L. Ren, "A compliant modular robotic hand with fabric force sensor for multiple versatile grasping modes," *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, no. June, 2016.

[7] K. C. Galloway, K. P. Becker, B. Phillips, J. Kirby, S. Licht, D. Tchernov, R. J. Wood, and D. F. Gruber, "Soft Robotic Grippers for Biological Sampling on Deep Reefs," *Soft Robotics*, vol. 3, no. 1, p. soro.2015.0019, 2016.

[8] L. U. Odhner and A. M. Dollar, "The smooth curvature model: An efficient representation of euler–bernoulli flexures as robot joints," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 761–772, aug 2012.

[9] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, jun 2010.

[10] A. D. Marchese, K. Komorowski, C. D. Onal, and D. Rus, "Design and control of a soft and continuously deformable 2d robotic manipulation system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2014.

[11] A. D. Marchese, R. K. Katzschmann, and D. Rus, "Whole arm planning for a soft and highly compliant 2d robotic manipulator," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014.

[12] A. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 840–869, oct 2015.

[13] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 1000–1019, 2016.

[14] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Dynamic control of soft robots interacting with the environment," *Arxiv Pre-Print*, 2018.

C Mathematical Detail on Smooth Curvature Model

C.1 Kinetic energy of a segment

The kinetic energy of a moving segment can be computed as

$$T = \frac{1}{2} m (\dot{x}_{cm} + \dot{y}_{cm})^2$$

using the center of mass coordinates of a segment as specified below.

$$x_{cm} = \frac{\int_0^L x(s) ds}{\int_0^L ds} = \frac{\int_0^L \int_0^s \cos(\phi(q, \hat{s})) d\hat{s} ds}{L}$$

$$y_{cm} = \frac{\int_0^L y(s) ds}{\int_0^L ds} = \frac{\int_0^L \int_0^s \sin(\phi(q, \hat{s})) d\hat{s} ds}{L}$$

C.2 Euler-Lagrangian: details and derivatives

These analytical derivatives are provided to be used in the optimizations

$$\frac{\partial \mathcal{L}}{\partial \alpha_1} + F_c \begin{bmatrix} \frac{\partial x_c}{\partial \alpha_1} \\ \frac{\partial y_c}{\partial \alpha_1} \end{bmatrix}^T \begin{bmatrix} \sin(\phi_c) \\ -\cos(\phi_c) \end{bmatrix} + N_a \begin{bmatrix} \frac{\partial x_t}{\partial \alpha_1} \\ \frac{\partial y_t}{\partial \alpha_1} \end{bmatrix}^T \begin{bmatrix} \cos(\phi_t) \\ \sin(\phi_t) \end{bmatrix} + \frac{\partial \phi_t}{\partial \alpha_1} M_a = 0$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_2} + F_c \begin{bmatrix} \frac{\partial x_c}{\partial \alpha_2} \\ \frac{\partial y_c}{\partial \alpha_2} \end{bmatrix}^T \begin{bmatrix} \sin(\phi_c) \\ -\cos(\phi_c) \end{bmatrix} + N_a \begin{bmatrix} \frac{\partial x_t}{\partial \alpha_2} \\ \frac{\partial y_t}{\partial \alpha_2} \end{bmatrix}^T \begin{bmatrix} \cos(\phi_t) \\ \sin(\phi_t) \end{bmatrix} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_3} + F_c \begin{bmatrix} \frac{\partial x_c}{\partial \alpha_3} \\ \frac{\partial y_c}{\partial \alpha_3} \end{bmatrix}^T \begin{bmatrix} \sin(\phi_c) \\ -\cos(\phi_c) \end{bmatrix} + N_a \begin{bmatrix} \frac{\partial x_t}{\partial \alpha_3} \\ \frac{\partial y_t}{\partial \alpha_3} \end{bmatrix}^T \begin{bmatrix} \cos(\phi_t) \\ \sin(\phi_t) \end{bmatrix} = 0$$

$$\frac{\partial \mathcal{L}}{\partial L} + F_c \begin{bmatrix} \frac{\partial x_c}{\partial L} \\ \frac{\partial y_c}{\partial L} \end{bmatrix}^T \begin{bmatrix} \sin(\phi_c) \\ -\cos(\phi_c) \end{bmatrix} + N_a \begin{bmatrix} \frac{\partial x_t}{\partial L} \\ \frac{\partial y_t}{\partial L} \end{bmatrix}^T \begin{bmatrix} \cos(\phi_t) \\ \sin(\phi_t) \end{bmatrix} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_1} = -\frac{\alpha_1 EI}{L}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_1^2} = -\frac{EI}{L}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_2} = -\frac{\alpha_2 EI}{3L}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_2^2} = -\frac{EI}{3L}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_3} = -\frac{\alpha_3 EI}{5L}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_3^2} = -\frac{EI}{5L}$$

$$\frac{\partial \mathcal{L}}{\partial L} = \frac{EI}{2L^2}\left(\alpha_1^2 + \frac{\alpha_2^2}{3} + \frac{\alpha_3^2}{5}\right) + EA_{eff}(L_0 - L)$$

$$\frac{\partial^2 \mathcal{L}}{\partial L^2} = -\frac{EI}{4L^3}\left(\alpha_1^2 + \frac{\alpha_2^2}{3} + \frac{\alpha_3^2}{5}\right) - EA_{eff}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_1 \partial L} = \frac{EI\alpha_1}{L^2}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_2 \partial L} = \frac{EI\alpha_2}{3L^2}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_3 \partial L} = \frac{EI\alpha_3}{5L^2}$$

C.3 State space representation and derivatives

$$\phi(q,s) = \alpha_1 \frac{s}{L} + \alpha_2 \left(\frac{s^2}{L^2} - \frac{s}{L}\right) + \alpha_3 \left(\frac{2s^3}{L^3} - \frac{3s^2}{L^2} + \frac{s}{L}\right)$$

tip angle $\phi_t(q)$:

$$\phi_t(q) = \phi_t(q, L) = \alpha_1$$

global point coordinates of point on beam at $s = s_c = cL$ with $0 \leq c \leq 1$

$$x_c(q) = \int_0^{s_c} \cos(\phi(q,s))ds = \int_0^{cL} \cos(\phi(q,s))ds$$

$$y_c(q) = \int_0^{s_c} \sin(\phi(q,s))ds = \int_0^{cL} \sin(\phi(q,s))ds$$

$$\phi_c(q) = \phi(q, cL) = \alpha_1 c + \alpha_2 \left(c^2 - c\right) + \alpha_3 \left(2c^3 - 3c^2 + c\right)$$

$$\frac{\partial \phi_c}{\partial c} = \alpha_1 + \alpha_2 \left(2c - 1\right) + \alpha_3 \left(6c^2 - 6c + 1\right)$$

$$\frac{\partial}{\partial \alpha_1}\phi_t = 1$$

$$\frac{\partial}{\partial \alpha_1}\phi = \frac{s}{L}$$

$$\frac{\partial}{\partial \alpha_2}\phi = \left(\frac{s^2}{L^2} - \frac{s}{L}\right)$$

$$\frac{\partial}{\partial \alpha_3}\phi = \left(\frac{2s^3}{L^3} - \frac{3s^2}{L^2} + \frac{s}{L}\right)$$

$$\frac{\partial^2}{\partial \alpha_i \partial \alpha_j}\phi = 0$$

$$\frac{\partial^2}{\partial \alpha_1 \partial L}\phi = -\frac{s}{L^2}$$

$$\frac{\partial^2}{\partial \alpha_2 \partial L}\phi = \frac{s}{L^2} - \frac{2s^2}{L^3}$$

$$\frac{\partial^2}{\partial \alpha_2 \partial L}\phi = -\frac{s}{L^2} + \frac{2s^2}{L^3} - \frac{6s^3}{L^4}$$

$$\frac{\partial}{\partial L}\phi = -\frac{s}{L^2}(\alpha_1 - \alpha_2 + \alpha_3) - \frac{2s^2}{L^3}(\alpha_2 - 3\alpha_3) - \frac{6s^3}{L^4}\alpha_3$$

$$\frac{\partial}{\partial \alpha_i}x_c = \int_0^{cL} -\sin(\phi)\frac{\partial \phi}{\partial \alpha_i}ds$$

$$\frac{\partial}{\partial \alpha_i}y_c = \int_0^{cL} \cos(\phi)\frac{\partial \phi}{\partial \alpha_i}ds$$

$$\frac{\partial}{\partial c}\frac{\partial x_c}{\partial \alpha_i} = -L\sin(\phi(cL))\frac{\partial \phi(cL)}{\partial \alpha_i}$$

$$\frac{\partial}{\partial c}\frac{\partial y_c}{\partial \alpha_i} = L\cos(\phi(cL))\frac{\partial \phi(cL)}{\partial \alpha_i}$$

using the Leibnitz rule:

$$\begin{aligned}\frac{\partial}{\partial L}x_c &= \frac{\partial}{\partial L}\int_0^{cL}\cos(\phi(q,s))ds \\ &= \cos(\phi(q,cL))c \\ &+ \int_0^{cL}\frac{\partial}{\partial L}\cos(\phi(q,s))ds \\ &= \cos(\phi(q,cL))c \\ &+ \int_0^{cL} -\sin(\phi)\frac{\partial \phi}{\partial L}ds\end{aligned} \tag{4}$$

and similarly

$$\begin{aligned}\frac{\partial}{\partial L}y_c &= \frac{\partial}{\partial L}\int_0^{cL}\sin(\phi(q,s))ds \\ &= \sin(\phi(q,cL))c \\ &+ \int_0^{cL}\cos(\phi)\frac{\partial \phi}{\partial L}ds\end{aligned} \tag{5}$$

$$\begin{aligned}\frac{\partial}{\partial c}\frac{\partial x_c}{\partial L} &= \cos(\phi_c) \\ &- c\sin(\phi_c)\frac{\partial \phi_c}{\partial c} \\ &- L\sin(\phi(s))\frac{\partial \phi(s)}{\partial L}\Big|_{s=cL}\end{aligned} \tag{6}$$