

Soft IK with Stiffness Control

James M. Bern and Daniela Rus

Abstract—Building robots using soft materials comes with numerous advantages, including inherent safety and natural compliance with the environment. However, it is quite challenging to match the strength and precision of rigid robots. In this work we ask how to get the best of both worlds, and make soft robots capable of acting stiff on demand. This would expand the capabilities of soft robots, and work towards the vision of using them to accomplish complex and variable tasks. We extend the open loop control method called Soft IK (soft robot inverse kinematics) to simultaneously control a cable-driven soft robot’s stiffness along with its deformed shape. Our method is an interactive multi-objective optimization, which reasons about stiffness by considering an ensemble of load cases. We show that by harnessing redundant actuation, our method is able to hit IK targets with controllable stiffness.

Index Terms—Modeling, Control, and Learning for Soft Robots

I. INTRODUCTION

Soft robots are inspired by animals, which use soft tissues to move with fluidity and robustness. And thanks to their softness, soft robots promise to go places rigid robots cannot [1]. With softness comes safety, meaning soft robots are well-suited to interact closely with people [2]. Also, because soft robots comply to their environment, they have an intrinsic ability to deal with physical uncertainty [3].

However stiffness has advantages as well. Traditional rigid robots are capable of great strength and precision. In this work we ask how to get the best of both worlds, and endow soft robots with the ability to modulate their stiffness. We envision robots able to act soft or stiff, depending on the needs of the situation. This mirrors what we see in nature, where humans and animals modulate their stiffness during locomotion and manipulation [4], [5]. The added versatility of controllable stiffness could make soft robots more suited to complex real-world tasks—such as home care and feeding assistance—which require an evolving mixture of traits. Certain subtasks demand strength and precision, while others necessitate prioritizing safety and compliance.

In pursuit of this goal, we focus on the problem of controlling the stiffness of cable-driven soft robots. Past work on Soft IK (soft robot inverse kinematics) has shown how differentiable simulation can be used to do intuitive open loop control of a soft robot’s deformed shape [6]. In this work we go a step further, and propose a computational method to do control soft robot shape and position. This method harnesses redundant actuation to simultaneously control a soft robot’s

stiffness along with its shape. The method takes the form a multi-objective optimization, which uses an ensemble of load cases to reason about stiffness within the framework of Soft IK. By leveraging differentiable simulation, we are able to navigate the complex interplay of soft materials, robot geometry, and cable actuation.

Our use of redundant cables is just one of a variety of actuation strategies that have been explored to change the stiffness of a soft robot [7], [8], though antagonistic actuation is common theme in soft robot stiffness modulation. There is work on using antagonistic activation of shape memory alloy (SMA) springs [9], multiple pneumatic muscle actuators (pMAs) in parallel [10], as well as single pMAs with multiple antagonistic chambers [11]. Hybrid strategies also exist, like using pneumatics to control shape and cables to control stiffness [12], [13]. Often these strategies go hand in hand with a particular soft robot design. A distinguishing feature of our work is that it is a method for controlling the stiffness of an arbitrary cable-driven soft robot, and can be used to augment the abilities of an existing robot.

Control strategies for soft robot stiffness modulation typically make strong assumptions about robot morphology. Works exist to control the stiffness of N -link soft robot arms [14], control the stiffness of soft robots amenable to being modeled with Cosserat rods [15], and to do impedance control for soft robots that follow a Piecewise Constant Curvature (PCC) model [16]. There are also works which make fewer assumptions about the robot, using more general simulators, or even model-free techniques including reinforcement learning [13]. We are inspired by the work of Koehler et al. on using FEM-based simulation to do soft robot impedance control [17]. We solve our control problem quite differently, performing an unconstrained nested optimization, where the inner optimization solves the robot’s full nonlinear physics. We also leverage redundant actuation. Our method is limited to open loop control, but is also quite general—applicable to any soft robot that can be modeled with our FEM-based differentiable simulator. We contribute:

- An extension of Soft IK to simultaneously control shape and stiffness of arbitrary cable-driven soft robots.
- Multiple simulated examples demonstrating our method.

II. BACKGROUND: FEM-BASED DIFFERENTIABLE SIMULATION OF SOFT ROBOTS

We employ a differentiable simulator based on the finite element method (FEM). This same general approach can be used to model cable-driven soft robots [6], [18] as well as the manipulation of elastic objects [19], [20]. In this section we will concisely summarize the simulator as it is used in

this paper, covering the general formulation, soft robot body, cables, fixed boundary conditions, and external forces. For more details, as well as how to model dynamics and basic contact please see [18].

A. General Formulation

Our simulator is based around a finite element mesh. We assemble the current position of each node in the mesh into a vector $\mathbf{x}$, which we refer to as the robot's *deformed shape*. Given some choice of *control input* $\mathbf{u}$, we call the resulting statically stable shape of the mesh $\mathbf{x}(\mathbf{u})$. If we assemble the total force acting on each node into a vector $\mathbf{F}$, then

$$\mathbf{x}(\mathbf{u}) = \underset{\mathbf{x}}{\text{solve}} \{ \mathbf{F}(\mathbf{u}, \mathbf{x}) = \mathbf{0} \}, \quad (1)$$

One option for finding $\mathbf{x}(\mathbf{u})$ is to solve the above equation directly using Newton's method, performing line search on e.g. $\|\mathbf{F}\|^2$. However if all forces are conservative—as is the case in this paper—we prefer instead to minimize the total energy of the system E . We solve

$$\mathbf{x}(\mathbf{u}) = \arg \min_{\mathbf{x}} E(\mathbf{u}, \mathbf{x}) \quad (2)$$

using Newton's method, performing line search on E . This is equivalent because all forces being conservative means $\mathbf{F} = -\frac{\partial E}{\partial \mathbf{x}}$, which implies a minimizer of E also solves static equilibrium.

B. Finite elements

We model the body of a soft robot as a finite element mesh, visible in Figure 1. We assemble the current position of each node into a vector $\mathbf{x}$, which we refer to as the robot's *deformed shape*. We use linear finite elements with a nonlinear neo-Hookean material model to model the body of the soft robot. On a practical level, these elements act like generalized springs, mapping from deformed shape $\mathbf{x}$ to the energy stored in the element.

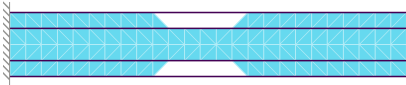


Fig. 1: Here we show a redundantly actuated 2D soft robot in its rest configuration. The robot is shaped like a dog bone, with four cables going down its length. The body of the robot is modeled as a finite element mesh (drawn in blue), and its cables are modeled as unilateral springs (drawn in dark purple). The robot is affixed to the left wall. In this paper we will show how to do simultaneous open loop position and stiffness control for cable-driven soft robots like this one.

C. Cables

The robot is actuated with cables (see Figure 2), which we consider as a polygonal chain running through via points in the mesh. The control inputs $\mathbf{u}$ are the *cable contractions*. For a given cable, u is the signed length of cable pulled onto (or let off of) its spool. To convert from cable contraction u to real-world motor angle θ , we use the relationship $u = R\theta$, where R is spool radius. Each cable is modeled as a unilateral

spring, mapping control input u and deformed shape $\mathbf{x}$ to stored potential energy. The cable stores zero energy when slack, and follows Hooke's law when taut.

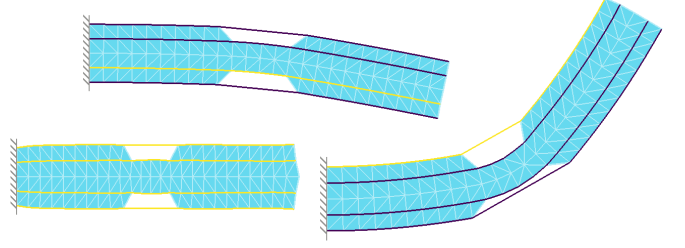


Fig. 2: We actuate the robot by winding its cables onto motorized spools. In this figure we show three statically stable poses $\mathbf{x}(\mathbf{u})$ resulting from different choices of control inputs $\mathbf{u}$. Taught cables are drawn in yellow, and slack cables in dark purple. A totally slack cable has no influence on the physics, e.g. for the rightmost pose, the lower three cables “might as well not be there.”

D. Boundary Conditions

The simulated robot can be affixed to a wall or floor using zero length springs. These can be seen as soft constraints on mesh position (in fact, so can the cable energies). Our avoidance of hard constraints enables us to use an unconstrained optimizer to solve physics.

E. External forces

In this work we apply constant external forces to the robot to reason about its stiffness. If we call the vector of external forces $\mathbf{F}_{\text{ext}}$, then they contribute potential energy equal to $-\mathbf{F}_{\text{ext}}^T \mathbf{x}$. This is the same way gravity is treated in [18].

III. BACKGROUND: SOFT IK

Soft IK is a general model-based method for doing open loop posing of soft robots (see Figure 3). It is analogous to traditional inverse kinematics for rigid robots [6]. Much like traditional IK, the goal of Soft IK is to find a control $\mathbf{u}^*$ that brings the robot's resulting deformed shape $\mathbf{x}(\mathbf{u})$ as close as possible to a user-specified target shape $\mathbf{x}'$. Soft IK works by using gradient descent to minimize the IK objective

$$\mathcal{O}_{\text{IK}}(\mathbf{u}) = \frac{1}{2} \|\mathbf{x}(\mathbf{u}) - \mathbf{x}'\|^2, \quad (3)$$

yielding optimal control inputs

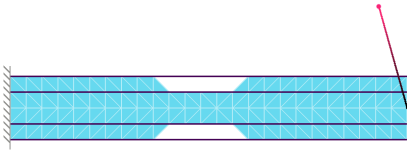
$$\mathbf{u}^* = \arg \min_{\mathbf{u}} \mathcal{O}_{\text{IK}}. \quad (4)$$

A. Control nodes

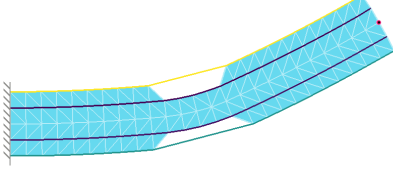
In practice we only prescribe target positions not for every single node in the mesh, but rather only for one or more user-defined *control nodes*. We assemble the positions of the control nodes into a vector $\mathbf{y}$, and have the IK objective

$$\mathcal{O}_{\text{IK}}(\mathbf{u}) = \frac{1}{2} \|\mathbf{y}(\mathbf{u}) - \mathbf{y}'\|^2. \quad (5)$$

Formally we might write $\mathbf{y} = \mathbf{S}\mathbf{x}$, where $\mathbf{S}$ is the appropriate selection matrix. In this paper $\mathbf{y}$ always refers to tip position.



(a) The user specifies a target tip position $\mathbf{y}'$ (drawn as a small red dot).



(b) Soft IK finds optimal control inputs $\mathbf{u}^*$ such that the resulting statically stable tip position $\mathbf{y}(\mathbf{u}^*)$ (drawn as the smaller black dot) matches the target.

Fig. 3: Summary of the original Soft IK method for controlling a soft robot's deformed shape. In this work we will extend Soft IK to simultaneously control stiffness.

B. Gradient computation

Soft IK works by minimizing $\mathcal{O}_{\text{IK}}$ using gradient descent. However to evaluate the gradient of our objective

$$\frac{d\mathcal{O}_{\text{IK}}}{d\mathbf{u}} = \frac{\partial\mathcal{O}_{\text{IK}}}{\partial\mathbf{u}} + \frac{\partial\mathcal{O}_{\text{IK}}}{\partial\mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{u}} \quad (6)$$

we must contend with the fact that Soft IK is a nested (bilevel) optimization. The function $\mathbf{x}(\mathbf{u})$ has no closed form expression; it is the result of solving the physical constraint in Equation (1). So to find $\frac{d\mathbf{x}}{d\mathbf{u}}$ we cannot just differentiate $\mathbf{x}(\mathbf{u})$ directly. In principle we could use numerical differentiation (finite differences) or automatic differentiation (and differentiate the entire routine of solving Equation (2) using Newton's method). However these approaches are liable to be slow, inaccurate, or both. Instead we leverage one of a family of techniques called *sensitivity analysis* [21].

1) *Direct Sensitivity Analysis*: Recall that $\mathbf{x}(\mathbf{u})$ solves the physical constraint $\mathbf{F}(\mathbf{u}, \mathbf{x}(\mathbf{u})) = \mathbf{0}$. The total derivative

$$\frac{d\mathbf{F}}{d\mathbf{u}} = \frac{\partial\mathbf{F}}{\partial\mathbf{u}} + \frac{\partial\mathbf{F}}{\partial\mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{u}} = \mathbf{0} \quad (7)$$

is therefore also zero. We compute the partial derivative terms analytically, and solve this system for $\frac{d\mathbf{x}}{d\mathbf{u}}$.

2) *Adjoint Sensitivity Analysis*: Direct sensitivity analysis requires solving Equation (7), which is a matrix-matrix equation. This can be slow, especially if the number of cables is large. A more efficient alternative is *adjoint sensitivity analysis* (the adjoint method), which works as follows. Equation (7) has $\frac{d\mathbf{x}}{d\mathbf{u}} = -\frac{\partial\mathbf{F}}{\partial\mathbf{x}}^{-1} \frac{\partial\mathbf{F}}{\partial\mathbf{u}}$, which we substitute into Equation (6) to reexpress the gradient as

$$\frac{d\mathcal{O}}{d\mathbf{u}} = \frac{\partial\mathcal{O}}{\partial\mathbf{u}} - \boldsymbol{\lambda}^T \frac{\partial\mathbf{F}}{\partial\mathbf{u}}, \quad (8)$$

where we have defined $\boldsymbol{\lambda}^T = \frac{\partial\mathcal{O}}{\partial\mathbf{x}} \frac{\partial\mathbf{F}}{\partial\mathbf{x}}^{-1}$. We find $\boldsymbol{\lambda}$ by solving the matrix-vector equation

$$\frac{\partial\mathbf{F}^T}{\partial\mathbf{x}} \boldsymbol{\lambda} = \frac{\partial\mathcal{O}^T}{\partial\mathbf{x}}. \quad (9)$$

C. Regularization

Soft IK benefits from two basic regularizers: $\|\mathbf{u}\|^2$ discourages large contractions, and $\|\mathbf{u} - \mathbf{u}_{\text{prev}}\|^2$ discourages taking large steps, where $\mathbf{u}_{\text{prev}}$ is the result of the previous step of gradient descent. Each can be scaled by a small positive constant and added to the objective.

D. Slack Handling

Slack cables are problematic for optimizations like Soft IK, because they create a flat region in the optimization landscape. I.e. if the i -th cable is slack, then $\frac{d\mathbf{x}}{du_i} = \mathbf{0}$, which implies $\frac{d\mathcal{O}}{du_i} = 0$. Therefore once a cable becomes slack, the optimizer will stop trying to use it (unless it happens to become taught as a side-effect of contracting other cables). One solution to this problem is to use a slack-eating regularizer [18]. Another is to add a projection step to the optimization, e.g. setting $u_i \leftarrow u_i + \max(10^{-4}, s_i)$, where s_i is the slack in the i -th cable (zero if none present).

IV. METHOD: SOFT IK WITH STIFFNESS CONTROL

The previously presented Soft IK method can only control deformed shape; it has no way of reasoning about stiffness. However, the soft robots we consider are redundantly actuated. I.e., for a given IK target, there exist not just one solution to the vanilla Soft IK problem, but rather a family of solutions. Our key observation is that while these solutions all reach the target equally well, they do so with different effective stiffness. Therefore by carefully choosing how much to pull the robot's redundant cables, we can control not just the robot's tip position, but also its stiffness.

In this section we integrate a means of controlling stiffness into the overall Soft IK framework, enabling simultaneous shape and stiffness control. As before, the IK goal will be to reach a target tip position. However we bake into our objective the following intuition: When acting stiff, a soft robot should resist physical disturbances, and remain close to the IK target. However when the robot is acting soft, physical disturbances should push it far away.

A. An Ensemble of Load Cases

Inspired by approaches in the field of topology optimization [22], we reason about stiffness by considering an ensemble of load conditions. Specifically we consider the original load condition with no external forces applied, and one or more user-specified load condition (e.g. pushing down on the tip of the robot with 50N). An example of this is shown in Figure 4. Note that except for external forces, we keep everything constant across conditions, including control input, rest shape of the mesh, cable layout, material parameters, etc.

Consider applying the same control input $\mathbf{u}$ in each load condition. Let $\mathbf{x}^{(0)}(\mathbf{u})$ denote the resulting deformed shape in the unloaded condition, and $\mathbf{x}^{(1)}(\mathbf{u}), \dots, \mathbf{x}^{(K)}(\mathbf{u})$ denote the resulting deformed shapes in the user-specified loaded conditions. We will use this super-scripting convention to denote the load condition of other quantities, e.g. $\mathbf{y}^{(0)}(\mathbf{u})$ is the resulting tip position in the unloaded condition.

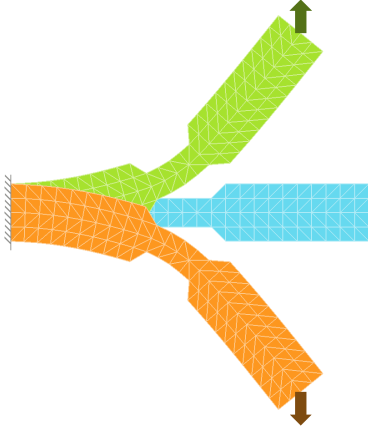


Fig. 4: We use an ensemble of load cases to reason about stiffness. In this figure we load the tip of the soft robot. The unloaded condition $\mathbf{x}^{(0)}(\mathbf{u})$ is shown in blue, the first loaded condition $\mathbf{x}^{(1)}(\mathbf{u})$ with 50N upward force is shown in green, and the second loaded condition $\mathbf{x}^{(2)}(\mathbf{u})$ with 50N downward force is shown in orange. Note that in this figure we have omitted the cables for visual clarity.

B. Control Objective

Our objective is a combination of an IK objective and a sum of stiffness objectives. We use a standard IK objective

$$\mathcal{O}_{\text{IK}}(\mathbf{u}) = \frac{1}{2} \left\| \mathbf{y}^{(0)}(\mathbf{u}) - \mathbf{y}' \right\|^2, \quad (10)$$

to quantify the mismatch between the unloaded tip position $\mathbf{y}^{(0)}(\mathbf{u})$ and target tip position $\mathbf{y}'$. We define a stiffness objective for each loaded condition

$$\mathcal{O}_{\text{Stiffness}}^{(i)}(\mathbf{u}) = \frac{1}{2} \left\| \mathbf{y}^{(i)}(\mathbf{u}) - \mathbf{y}^{(0)}(\mathbf{u}) \right\|^2, \quad (11)$$

which is the squared distance between loaded and unloaded tip positions $\mathbf{y}^{(i)}(\mathbf{u})$ and $\mathbf{y}^{(0)}(\mathbf{u})$ respectively. Intuitively, this captures how far away the load has managed to push the robot from the unloaded position.

For a stiff-acting robot we would expect the sum of the stiffness objectives to be small—since it resists external loads—and for a soft-acting robot we would expect it to be large. The weighted sum of the IK objective and the stiffness objectives forms the overall objective function for Soft IK with stiffness control

$$\mathcal{O}_{\text{IK+Stiffness}}(\mathbf{u}) = \mathcal{O}_{\text{IK}}(\mathbf{u}) + \phi \sum_{i=1}^K \mathcal{O}_{\text{Stiffness}}^{(i)}(\mathbf{u}). \quad (12)$$

We use the parameter ϕ to find a control of desired stiffness. First note that by setting $\phi = 0$ we recover the original Soft IK objective, which has no way of reasoning about stiffness. We can choose $\phi > 0$ to look for a *stiffer control*—one that keeps the robot close to the target even when subjected to physical disturbances. And we can choose $\phi < 0$ to look for a *softer control*—one that lets the robot be pushed away from the target by physical disturbances.

C. Gradient Computation

In our method the different conditions are all simulated separately, so gradient computation can be done by applying the method in Section III-B for each condition. For example, to compute the gradient using direct sensitivity analysis, we would calculate the Jacobians $\frac{d\mathbf{x}^{(0)}}{d\mathbf{u}}, \frac{d\mathbf{x}^{(1)}}{d\mathbf{u}}, \dots, \frac{d\mathbf{x}^{(K)}}{d\mathbf{u}}$ by solving a system like the one in Equation (7) for each load condition.

As a final conceptual note, observe that similarly to what is done for trajectories in [18], if we assemble the resulting mesh shapes from all load conditions into a vector

$$\mathbf{x}(\mathbf{u}) = \left[\mathbf{x}^{(0)}(\mathbf{u}) \quad \mathbf{x}^{(1)}(\mathbf{u}) \quad \dots \quad \mathbf{x}^{(K)}(\mathbf{u}) \right]^T, \quad (13)$$

then we can write all physical constraints as a single equation

$$\underbrace{\begin{bmatrix} \mathbf{F}^{(0)}(\mathbf{u}, \mathbf{x}^{(0)}(\mathbf{u})) \\ \mathbf{F}^{(1)}(\mathbf{u}, \mathbf{x}^{(1)}(\mathbf{u})) \\ \vdots \\ \mathbf{F}^{(K)}(\mathbf{u}, \mathbf{x}^{(K)}(\mathbf{u})) \end{bmatrix}}_{\mathbf{F}(\mathbf{u}, \mathbf{x}(\mathbf{u}))} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\mathbf{0}}, \quad (14)$$

which is of the form

$$\mathbf{F}(\mathbf{u}, \mathbf{x}(\mathbf{u})) = \mathbf{0}. \quad (15)$$

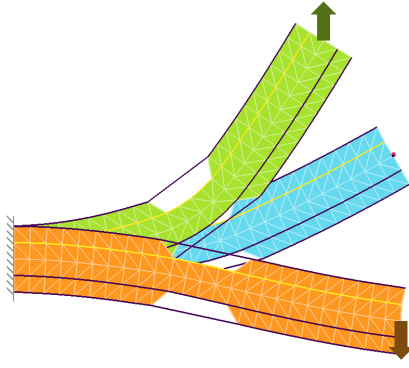
Recall that this is the starting point for sensitivity analysis, and observe that solving for the Jacobians $\frac{d\mathbf{x}}{d\mathbf{u}}, \frac{d\mathbf{x}^{(1)}}{d\mathbf{u}}, \dots, \frac{d\mathbf{x}^{(K)}}{d\mathbf{u}}$ separately is equivalent to solving the single larger system $\frac{\partial \mathbf{F}}{\partial \mathbf{u}} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{u}} = \mathbf{0}$ all at once, given that $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ is block diagonal.

D. Slack Handling

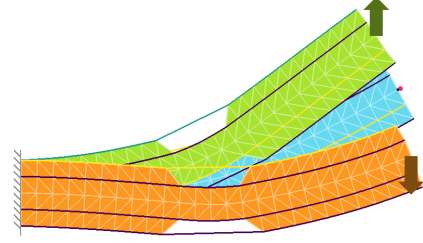
We use the projection step described in Section III-D, with one key modification: For each cable we modify $\mathbf{u}$ so that the *minimum slack across load conditions* is slightly positive. This ensures each cable is taught in at least one load condition, and therefore visible to the optimizer. To understand why this scheme is preferable to e.g. removing all slack from all load conditions, note that optimal control inputs often *require slack*. For example, consider the top left image in Figure 7. The goal is to reach the target while acting soft. The method allows a generous amount of slack in the unloaded condition (blue), which enables the robot to deform freely in the loaded conditions (green and orange). If there were no slack present in the unloaded condition, then the robot would not act nearly so soft when loaded.

V. EXAMPLES

We use our method to control the position and stiffness of multiple robots in simulation. For footage of simulations please see our video.



(a) Softer control ($\phi < 0$)



(b) Stiffer control ($\phi > 0$)

Fig. 5: Soft IK with stiffness control lets us simultaneously satisfy an IK objective and control stiffness. Here we have used the optimizer to find softer and stiffer controls for the same target tip position $\mathbf{y}'$ (drawn as the small red dot). Note how both solutions hit the target in the unloaded conditions (blue), but differ greatly in how much they resist external forces in the loaded conditions (green and orange). The softer control contracts a single inner cable, and releases slack in the outer cables, leaving the bar free to bend in its slender middle region. The stiffer control cocontracts multiple cables, in order to resist bending. Note that in the stiffer case the user has traded a small amount of positional accuracy for additional stiffness.

A. Soft Beam

We control the four cable 2D soft beam shown throughout the paper. Our controller makes it possible to reach a target tip position with user-specified stiffness. We show examples of reaching the same target with softer and stiffer controls in Figures 5 and 7. To further understand the action of the different controls, we plot force-displacement curves in Figure 6. We observe that our chosen ensemble of load conditions yields a control that is significantly stiffer even for forces far beyond 50N. Additionally we note that the relationship between the two curves is not a simple one like pure scaling. As we increase the force from 0N to 50N the curves grow farther apart, but then they start coming back together. Future work might look at deliberately shaping this force response.

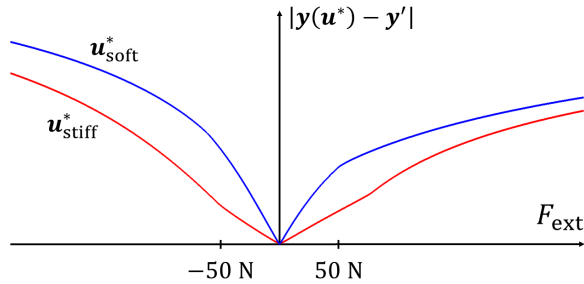


Fig. 6: Force-displacement curves for the control inputs pictured in Figure 5, $\mathbf{u}_{\text{soft}}^*$ in blue and $\mathbf{u}_{\text{stiff}}^*$ in red. These curves were created by applying external load F to the tip (negative force means pushing down, and positive force means pulling up), and then solving statics and evaluating displacement. We consider displacement as the distance (absolute error) between deformed and target tip positions $|\mathbf{y}(\mathbf{u}^*) - \mathbf{y}'|$.

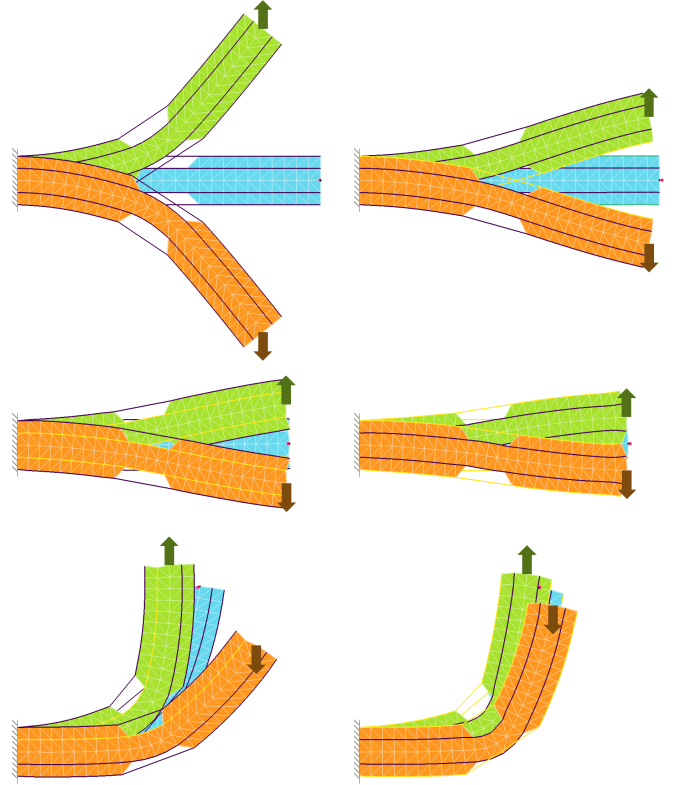
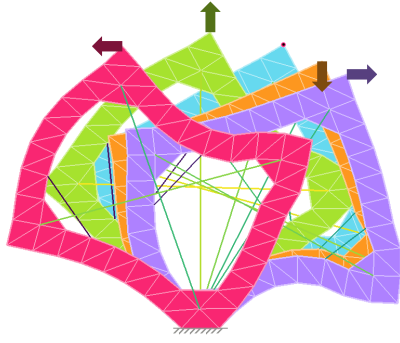
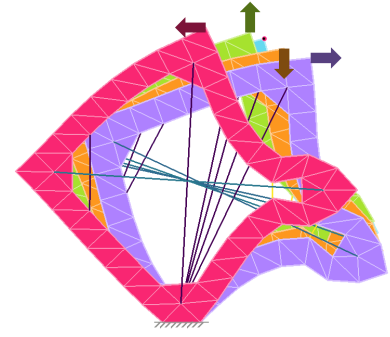


Fig. 7: Softer (left) and stiffer (right) controls found for various additional targets. Note that the achievable range of stiffness depends greatly on the target. We can achieve two very different stiffness for the target in top row, by letting out a great deal of slack (on the left), or taking it in (on the right). However to hit the target in middle row, the system must significantly compress the beam, which narrows the range of possible stiffnesses. As in Figure 5, we have in some cases traded small amounts of positional accuracy to match the desired stiffness.



(a) Softer control ($\phi < 0$)



(b) Stiffer control ($\phi > 0$)

Fig. 8: We use an ensemble of four load cases in our control optimization for this diamond-shaped robot. Here we show two choices of control inputs found for the same target tip position. Note that the stiffer control keeps the robot closer to the target in all of the loaded conditions.

B. Diamond-shaped Robot

We simulate a diamond-shaped robot, inspired by [23]. The robot is actuated by four cables, consisting of a shorter cable on either side, and longer cables running vertically and horizontally. To reason about the robot's stiffness we use an ensemble of four load conditions, consisting of 50N applied to the tip in each of the cardinal directions. Results are shown in Figures 8 and 9.

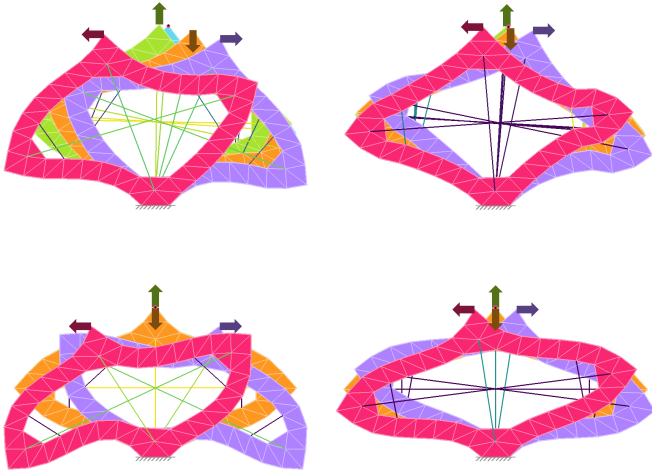


Fig. 9: Softer (left) and stiffer (right) controls found for additional targets. To reach the target in the bottom row, the system sacrifices the ability to control stiffness in the vertical.

VI. DISCUSSION AND FUTURE WORK

A. Designing for Controllable Stiffness

In this work we showed how to leverage redundant cables to control stiffness along with position. Stepping back however, we observe that stiffness is determined not just by cable contractions, but also by the robot's design—shape, cable layout, and materials parameters. These factors come together to produce the force displacement curves in Figure 6, as well as situations like in Figure 9, where we can control stiffness in one direction but not another.

If we are to wield fine-grained control over stiffness—to help attack the challenging problems of soft manipulation and locomotion—we must understand not only how to actuate our robots, but also how to design them. An exciting avenue of future work would be to the proposed method as the inner loop of a design optimization, similarly to how the original Soft IK method was used to help solve the problem of automated cable layout [24]. To give a sense of the range of possibilities that may be enabled by new computational tools, we design and simulate several more soft bars, shown in Figure 10 and in our video.

B. Tuning the Parameter ϕ

In this work we enable the user to control stiffness by tuning a single scalar parameter ϕ . This is a convenient choice, as it is generally simple and intuitive to scrub through values of ϕ in order to find a control with desired stiffness properties. However, it is not without its drawbacks. Small changes in the parameter ϕ can cause large changes in control u^* , and different spatial targets often require different values of ϕ to achieve control inputs with visually similar stiffness. These facts can at times make using our system somewhat cumbersome, as ϕ must be re-tuned for each successive target. We show this in our video. A solution might be to have the user instead specify the desired distance between loaded and unloaded tip positions.

C. Importance of Load Selection

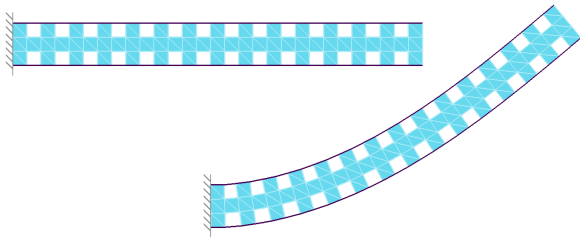
An important point is that our system controls stiffness *with respect to the user-defined load conditions*. Soft IK with stiffness control is an open loop method, where we rely on the user to select an ensemble of load conditions that efficiently spans the space of external loads the robot is liable to encounter when deployed the real world. Certain tasks will have greater uncertainty in this regard, especially if the robot must interact closely with people. In such cases, a general purpose set of load conditions (e.g. 50N in each of the cardinal directions, as done for the diamond-shaped robot) is likely advisable.

VII. CONCLUSION

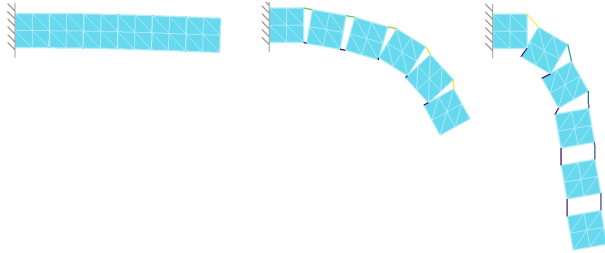
In this work we presented a model-based open loop control method for soft robot shape and stiffness, which builds a practical notion of stiffness into soft robotic inverse kinematics. We demonstrated our method on multiple simulated examples, accomplishing inverse kinematics tasks with softer and stiffer controls. Our results highlight avenues for further exploration, including harnessing computational design to exhibit more fine grained control over stiffness. By leveraging differentiable simulation, we are able to get the same soft robot to do more—and harness redundant cables to achieve multiple objectives at the same time.

REFERENCES

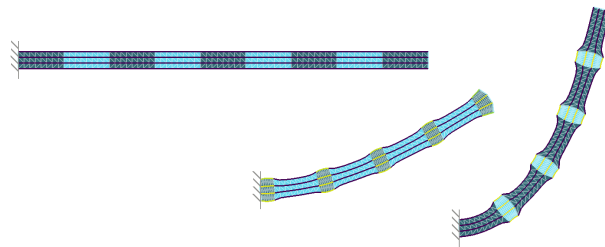
- [1] D. Rus and M. T. Tolley, “Design, fabrication and control of soft robots,” *Nature*, 2015.
- [2] T. Arnold and M. Scheutz, “The tactile ethics of soft robotics: Designing wisely for human–robot interaction,” *Soft robotics*, vol. 4, no. 2, pp. 81–87, 2017.



(a) A soft beam with many small cutouts running down its length. When subjected to external force it deforms smoothly along its length, unlike the example shown throughout this paper with the slender middle region.



(b) A soft beam made of multiple separate pieces, which are held together by cable tension. The cables can be released to yield a sudden, dramatic change in stiffness.



(c) A soft beam with nonuniform material parameters (softer regions in dark blue, stiffer regions in light blue). By tensing and releasing groups of cables running through either the different regions, the effective stiffness of the bar can be changed while preserving length.

Fig. 10: Multiple simulated concepts for a soft beam.

- [3] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, and H. M. Jaeger, “Universal robotic gripper based on the jamming of granular material,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 44, pp. 18 809–18 814, 2010.
- [4] H. Gomi and R. Osu, “Task-dependent viscoelasticity of human multijoint arm and its spatial characteristics for interaction with environments,” *Journal of neuroscience*, vol. 18, no. 21, pp. 8965–8978, 1998.
- [5] T. R. Clites, A. S. Arnold, N. M. Singh, E. Kline, H. Chen, C. Tugman, B. Billadeau, A. A. Biewener, and H. M. Herr, “Goats decrease hindlimb stiffness when walking over compliant surfaces,” *Journal of Experimental Biology*, vol. 222, no. 10, 2019.
- [6] J. M. Bern, G. Kumagai, and S. Coros, “Fabrication, modeling, and control of plush robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3739–3746.
- [7] M. Manti, V. Cacucciolo, and M. Cianchetti, “Stiffening in soft robotics: A review of the state of the art,” *IEEE Robotics & Automation Magazine*, vol. 23, no. 3, pp. 93–106, 2016.
- [8] Y. Yang, Y. Li, and Y. Chen, “Principles and methods for stiffness modulation in soft robot design and development,” *Bio-Design and Manufacturing*, vol. 1, no. 1, pp. 14–25, 2018.
- [9] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, “Soft robot arm inspired by the octopus,” *Advanced Robotics*, vol. 26, no. 7, pp. 709–727, 2012.
- [10] M. E. Giannaccini, C. Xiang, A. Atyabi, T. Theodoridis, S. Nefti-Meziani, and S. Davis, “Novel design of a soft lightweight pneumatic continuum robot arm with decoupled variable stiffness and positioning,” *Soft Robotics*, vol. 5, no. 1, pp. 54–70, 2018.
- [11] N. S. Usevitch, A. M. Okamura, and E. W. Hawkes, “Apam: antagonistic pneumatic artificial muscle,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1539–1546.
- [12] A. Shiva, A. Stilli, Y. Noh, A. Faragasso, I. D. Falco, G. Gerboni, M. Cianchetti, A. Menciassi, K. Althoefer, and H. A. Wurdemann, “Tendon-based stiffening for a pneumatically actuated soft manipulator,” *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 632–637, 2016.
- [13] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, and C. Laschi, “Multiobjective optimization for stiffness and position control in a soft robot arm module,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 108–115, 2018.
- [14] G. Tonietti and A. Bicchi, “Adaptive simultaneous position and stiffness control for a soft robot arm,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2. IEEE, 2002, pp. 1992–1997.
- [15] M. Mahvash and P. E. Dupont, “Stiffness control of surgical continuum manipulators,” *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 334–345, 2011.
- [16] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, “Dynamic control of soft robots interacting with the environment,” in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 46–53.
- [17] M. Koehler, A. M. Okamura, and C. Duriez, “Stiffness control of deformable robots using finite element modeling,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 469–476, 2019.
- [18] J. Bern, P. Banzet, R. Poranne, and S. Coros, “Trajectory optimization for cable-driven soft robot locomotion,” in *Proceedings of Robotics: Science and Systems*, Freiburg/Breisgau, Germany, June 2019.
- [19] S. Duenser, J. M. Bern, R. Poranne, and S. Coros, “Interactive robotic manipulation of elastic objects,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3476–3481.
- [20] S. Duenser, R. Poranne, B. Thomaszewski, and S. Coros, “Robocut: Hot-wire cutting with robot-controlled flexible rods,” *ACM Trans. Graph.*, vol. 39, no. 4, July 2020.
- [21] A. M. Bradley, “Pde-constrained optimization and the adjoint method,” 2010.
- [22] A. Diaz and M. Bendsøe, “Shape optimization of structures for multiple loading conditions using a homogenization method,” *Structural optimization*, vol. 4, no. 1, pp. 17–22, 1992.
- [23] C. Duriez, “Control of elastic soft robots based on real-time finite element method,” in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3982–3987.
- [24] J. M. Bern, K.-H. Chang, and S. Coros, “Interactive design of animated plushies,” *ACM Transactions on Graphics (TOG)*, 2017.