Model-based optimization of tendon layouts for soft robots

James M. Bern[1], Luca Lavarini, [1], Roi Poranne[1,2], and Stelian Coros[1]

***Abstract*— Soft robots are capable of a wide range of motions involving large deformations throughout their entire bodies. This flexibility enriches the space of possible motions such robots can create, promising to solve manipulation and locomotion problems in new and exciting ways [1]. In this work we focus specifically on tendon-actuated soft robots, like those in [2]. The space of motions such a robot can achieve is a highly nonlinear function of its tendon layout. In this work we attack the challenging inverse design problem of automatically generating physically-realizable tendon layouts capable for user-specified target motions.**

I. Introduction

Finding an optimal actuation layout for a tendon-driven soft robot is a challenging task. On one hand, more tendons enable finer control over the pose of the robot. On the other hand, they increase its cost and complexity. Our goal is to strike a balance between the number of tendons and the range of motions they permit. This work builds upon the method of [3], a two stage interactive approach consisting of a *soft IK* stage and a *network discovery* stage. Both stages hinge on on the idea of rigging an *idealized robot* with a dense set of small, independently-actuated, contractile elements called *fibers*. In the soft IK stage, the user interactively specifies a target pose. The system automatically computes fiber contractions that deform the idealized robot to match this target. Once the user is satisfied with the pose, the network discovery stage converts the most active fibers into a single network of continuous tendons. The process, in short, is a heuristic-driven graph search that builds a sparse set of tendons with contractions similar to those of the active fibers. These tendons are attached to the *realizable robot*, which is observed to achieve visually similar poses to those achieved by the idealized robot.

The soft IK approach proves to be quite versatile, and can be used to pose robots with hundreds of fibers or tendons. We will make use of it here as well. However, the overall two-stage approach of soft IK followed by network discovery has significant limitations. First, the realizable robot cannot always recreate the poses made using the idealized robot and soft IK. Second, the network discovery algorithm was found to perform poorly when the density of the finite element mesh and fiber layout were increased.

The first limitation is a direct result of the fact that soft IK and network discovery are split into two separate steps. The second limitation is a shortcoming of the graph search used, which gets lost in the presence of parallel strands of fibers with similar tension.

In this work we propose to eliminate the network discovery stage, and instead incorporate its *objective* into soft IK. To this end we craft regularizers that encourage the formation of sparse, continuous tendons. This approach overcomes the main limitations of the previous network discovery procedure, while retaining the expressive freedom of the soft IK posing system.

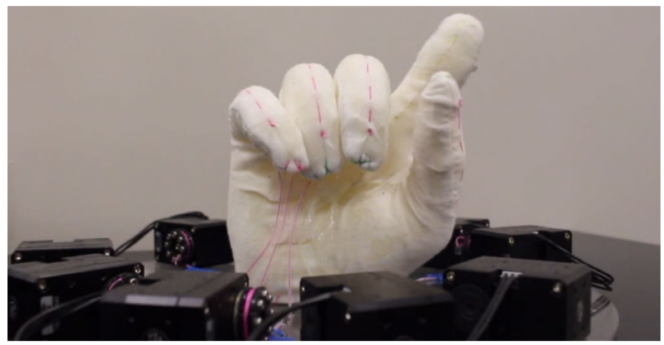

Fig. 1. A soft robotic hand with naive tendon routing. A more clever actuation layout could have likely achieved this pose with fewer tendons

II. Problem statement

For the sake of completeness, we briefly introduce the notation of [3], which we build upon. We represent the body of the soft robot as a finite element (FE) triangular mesh, and denote its initial node positions by $\mathbf{x}^0$. The dense fiber mesh is an independent graph embedded within the FE mesh, with node positions $\mathbf{c}(\mathbf{x}^0)$, represented as barycentric coordinates with respect to $\mathbf{x}^0$. This is in contrast to [3] who used the same mesh for both. Fibers can contract independently; a virtual gearmotor can pull each one, effectively changing its rest length. We let α^0 denote the initial rest length of a fiber, α^c the contraction due to the motor, and α the as the current rest length, $\alpha = \alpha^0 - \alpha^c$. The deformation of a fiber Γ, is the change in length with respect to the current rest length, and $U(\Gamma)$ is the unilateral strain energy as defined in [3]. Finally, $\tau = \frac{\partial U}{\partial \Gamma}$ is the tension on the fiber.

Given a deformed state with positions $\mathbf{x}$, The total deformation energy of the system is

$$E_{\text{tot}} = E_{\text{body}}(\mathbf{x}) + E_{\text{contr}}(\mathbf{c}(\mathbf{x}), \alpha^c) \tag{1}$$

where E_{body} is the (elastic) deformation energy of the soft body, and E_{contr} is the strain energy stored in the fibers, which depends on the contractions, α^c. In other words, the the contraction α^c determine the pose of the model, i.e. the

[1] Department of Computer Science, ETH Zurich, Switzerland. jamesmbern@gmail.com, sancierra@gmail.com, roi.poranne@inf.ethz.ch, scoros@gmail.com
[2] Department of Computer Science, University of Haifa, Israel

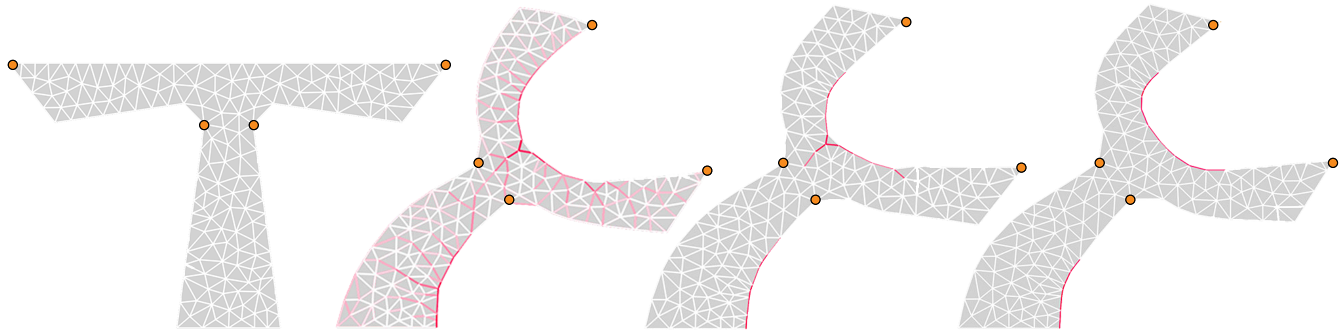

Fig. 2. Regularizing the tension in the T shape. (From left to right) The T shape in its undeformed state and the target deformation without regularization. Red segments are fibres that are tensioned. Without any regularization the activation pattern presents itself to be very dense. Applying an L_0 regularizer causes significant sparsification of activation. We further apply a smoothing regularizer to encourage continuous final tendons with the result being depicted on the right.

minimizer of (1). The goal in [3] is to find the *contractions* α^c that deforms the robot to a desired position. This is done by minimizing

$$\mathcal{O}_{\text{pos}} = \|\left(S\hat{\mathbf{x}} - \mathbf{x}'\right)\|^2, \tag{2}$$

where $\hat{\mathbf{x}}$ is the node positions at equilibrium, and S is a selector matrix that picks specific nodes that the user indicated should go to $\mathbf{x}'$. Noting that $\hat{\mathbf{x}}$ is a function of α^c, that is, $\hat{\mathbf{x}}(\alpha^c) = \arg\min_x E_{\text{tot}}(\mathbf{x}, \alpha^c)$, [3] propose to compute the gradient of $\mathcal{O}_{\text{pos}}$ with respect to α^c via the chain rule

$$\frac{d\mathcal{O}_{\text{pos}}}{d\alpha^c} = \frac{\partial \mathcal{O}_{\text{pos}}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\alpha^c} \tag{3}$$

and minimize (2) with a gradient based optimization method. An analytic expression for $\frac{d\mathbf{x}}{d\alpha^c}$ is found using sensitivity analysis. The approach works well, in the sense that the activation of the fibers gives the correct pose. However in practice we note two areas for improvement. First, many more fibers than necessary are used. Second, the amount tension can vary greatly between nearby fibers. Our goal in this work is to add a regularization term that promotes solutions with few, long strands of fibers of similar tensions.

A. Fiber regularization

We leverage the observation that a chain of equal-tension fibers, or *strand*, is physically-equivalent to a single tendon with the same tension. Our goal is to achieve poses using only a few such active strands of fibers. This would eliminate the need for network discovery, as each strand could simply be replaced with a corresponding tendon.

We first add a regularizer to (2) that rewards sparsity. However, this alone is not sufficient to satisfy our goal. Fibers with different tension, as well as gaps between strands of fibers, limit our ability to merge fibers into tendons. Additionally, we observe that tendons with sharp corners, i.e. "kinks," will incur large frictional cost when installed in real-world prototypes, and should be avoided unless absolutely necessary. To combat these issues we add a regularizer to (2) that rewards smoothness.

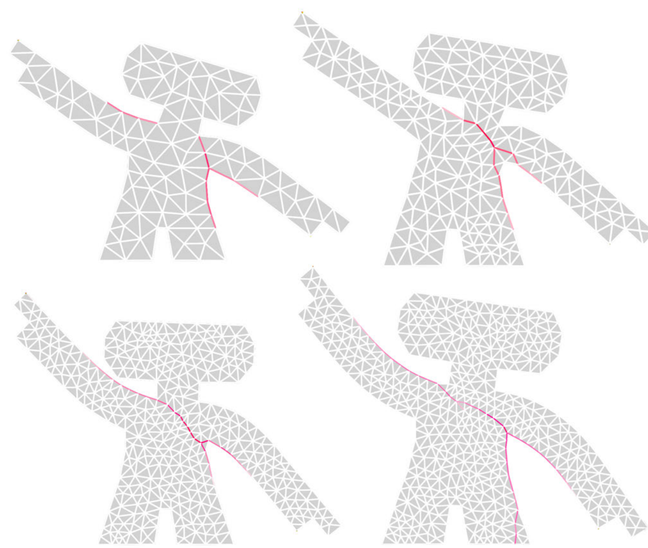

Fig. 3. Robot mesh with target pose and corresponding activation pattern. Both sparsity and smoothness regularization have been used. The calculation has been done on meshes with four different triangulation densities. For comparison see [3].

III. Fiber sparsifiction via L_0- regularization

We use an L_0 regularization to encourage the optimizer to use only a small number of the many fibers available to it. Consider,

$$\mathcal{O}_{\text{sparsity}}(\alpha^c) = \|\hat{\tau}(\alpha^c)\|_0, \tag{4}$$

This regularizer simply counts the number of fibers with positive tension. Unfortunately, an optimization problem with the regularizer (4) is usually intractable. The reason why is because the proposed regularizer is not continuous and normally requires combinatorial optimization, which is expected to be extremely costly for large number of fibers. To circumvent this issue, we use the *homotopy* method [4]: Rather than to optimize the L_0-norm directly, we instead replace it with a sequence of smooth functions, that approaches the L_0-norm in the limit. We propose using this simple parameterized sequence:

$$\mathcal{O}_{\text{sparsity}}(\alpha^c, k) = \begin{cases} 0,\ \tau(\alpha^c) < 0 \\ s(\tau(\alpha^c)),\ 0 \le \tau(\alpha^c) \le k \\ 1,\ k < \tau(\alpha^c) \end{cases}, \qquad (5)$$

where $s(\tau)$ is a cubic spline on $[0, k]$ that smoothly transitions from 0 to 1. We slowly *sharpen* this transition by decreasing k towards 0. After each decrease, we continue to run the optimizer until it settles, before decreasing k again. This process is repeated until there is no visible change in the activation. Fig. 2 shows the result of sparsifying the activation on a T shape.

IV. Continuous tendons via smoothing

We employ a *directional* smoothness regularizer to encourage the optimizer to find long, smooth and continuous strands of equi-tensioned fibers. It helps by filling small gaps (Fig. 4), reducing kinks and promoting directionality in the tendons (Fig. 6). We define the regularizer by

$$\mathcal{O}_{\text{continuity}} = \sum_{a,b} s_{ab}\big(\tau_a - \tau_b\big)^2 \qquad (6)$$

where the sum is over all pairs of fibers that share a node. The weight $s_{ab} = cos^2(\theta_{ab})$, θ_{ab} being the angle between two neighboring fibers in the undeformed state, ensures that only the differences in tension between fibers that have similar orientations is penalized. This weighting function additionally dampens the activation of all neighbouring fibres *perpendicular* to active fiber strands, which creates a cleaner result (see Fig. 5).

V. Future plans

We present preliminary results showing the effectiveness of our sparsity and continuity regularizers. Activation sparsification can be chosen to be arbitrarily strong and continuity is strenghtened by our smoothness regularizer. Target poses defined can still be fully and accurately achieved. Our final tendons are equal to or exceed the tendons resulting from the techniques used in [3] in terms of quality. Furthermore, quality does not degrade if we increase the fiber grid resolution of the finite element mesh (Fig. 3).

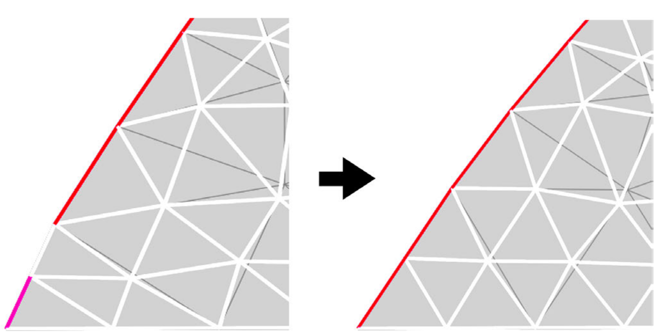

Fig. 4. The smoothness regularizer closes gaps in active fibre strands. In this case, $\mathcal{O}_{\text{continuity}}$ drives the activation gap up in tension to enforce equitension in the chain.

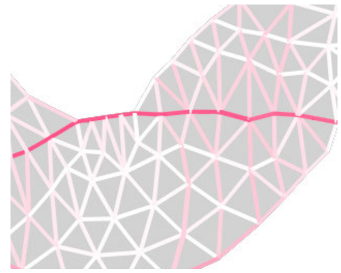

Fig. 5. The smoothing term dampens fibers perpendicular to main strands. Fibers wtill tend to generate a slight activation pattern around them if the smoothing objective has a lot of weight, but this effect can be countered by increasing the L_0 regularization.

Our current implementation is not quite efficient. In the future we plan to improve it, which will allow us to extend our approach to support optimization for multiple target poses simultaneously, 3D models, and user-in-the-loop optimization. Another component that is still missing is a regularization term which will directly ensure that fibers along a strand are equally tensioned. This is currently achievable to some small extent with the smoothness term, but we would be interested in a term that guarantees it.

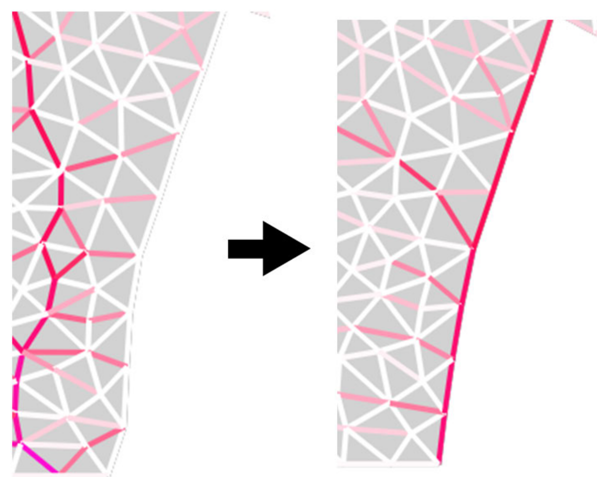

Fig. 6. Our objective is highly non-convex, especially in the presence of the sparsity regularization. $\mathcal{O}_{\text{continuity}}$ can be used to explore potentially better local minima. In this example, a messy activation patterncan be seen. Enabling $\mathcal{O}_{\text{continuity}}$ allows the solver to converge to a cheaper and smoother solution.

References

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

[2] J. M. Bern, G. Kumagai, and S. Coros, "Fabrication, modeling, and control of plush robots," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2017.

[3] J. M. Bern, K.-H. Chang, and S. Coros, "Interactive design of animated plushies," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 80, 2017.

[4] R. Poranne, M. Tarini, S. Huber, D. Panozzo, and O. Sorkine-Hornung, "Autocuts: Simultaneous distortion and cut optimization for uv mapping," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 215:1–215:11, Nov. 2017. [Online]. Available: http://doi.acm.org/10.1145/3130800.3130845