An Energy-Dense Two-Part Torsion Spring Architecture and Design Tool

Zachary Bons, *Graduate Student Member, IEEE*, Gray C. Thomas, *Member, IEEE*, Luke Mooney, and Elliott J. Rouse, *Senior Member, IEEE*

***Abstract*—Emerging wearable, assistive, and mobile robots seek to interact with the environment and/or humans in a compliant, dynamic, and adaptable way. Springs are critical to achieving this objective, but the associated increase in volume, mass, and complexity is limiting their application and impact in this rapidly developing field. This article presents a novel rotary spring architecture that is both lightweight and compact. Our two-part spring consists of radially-spaced cantilever beams that interface with an internal, gear-like camshaft. We present the concept and equations governing their mechanics and design. To facilitate broad adoption, we introduce an open-source design tool, which enables the design of custom springs in minutes instead of hours or days. We also empirically demonstrate our design with four test springs and validate the achievement of target spring rates and deflections. Finally, we present several redesigns of existing springs in the robotics literature to demonstrate the wide applicability of our spring architecture.**

***Index Terms*—Design, prosthetics, robotics, series elastic actuators (SEA), springs, wearable robotics.**

I. Introduction

SPRINGS are essential building blocks for many engineering applications, including storing elastic energy and measuring force or torque. This combination of functionality makes springs highly desirable in human-centered robotic applications (e.g., rehabilitation technologies, exoskeletons, and prostheses); however, the added weight and complexity of incorporating springs can result in a difficult tradeoff for these mobile systems [1], [2], [3], [4], [5]. One common implementation involves placing a spring between the output of the transmission and the load, which is known as a series elastic actuator (SEA) [6]. Although not without drawbacks—namely force bandwidth reduction and the aforementioned increase in mass and complexity—this design paradigm has notable benefits, including compliant interaction with the environment, torque feedback, energy storage, and improved shock tolerance. In recent years, the SEA application has become a driving force in torsion spring designs. For mobile robot systems, these designs have prioritized specific energy (energy storage per mass) and energy density (energy storage per volume). With SEAs becoming more prevalent, simple torsion spring designs, including thin-walled tubes and cantilever beams, pose challenges in packaging and volume. Torsion tube springs with thin walls [7], [8] and long beam flexures with cam-rollers [9] or hinges [10] can be effective for low stiffness applications, and their low weight results in high specific energy. However, to achieve appropriate stiffness values, the aspect ratio—length over diameter—must be considerable [11]. Thus, these approaches result in long, thin springs that are often impractical for rotary joints in robots. This challenge has resulted in an era of spring designs that prioritize convenient packaging in addition to specific energy and energy density.

To achieve convenient packaging, recent designs have emphasized disk-like architectures that balance specific energy, compactness, and required mechanics. These springs incorporate torsional elasticity between a central anchor point rotated with respect to an outer rim. Some early researchers achieved this design by arranging prismatic springs tangentially around a circular frame [12], [13]. However, this approach results in a nonlinear stiffness behavior [13] and has the added complexity of several removable parts. The latter concern may be responsible for the transition to monolithic springs in subsequent designs. One of the most common and successful planar monolith spring designs connects the central anchor to the rim with one or more spiral arms [14], [15], [16], [17], [18], [19]. This configuration is efficient and works well with a wide range of stiffness coefficients (e.g., 30–800 Nm/rad); however, the nature of the spiral arms often causes differences in stiffness depending on the direction of deflection [19]. Other designers have instead emphasized torque-sensing resolution [5], [20], [21], [22], [23], [24], [25] in their development. These designs have produced compact springs that are generally limited in their deflection

Manuscript received 26 July 2023; revised 27 September 2023; accepted 23 October 2023. Recommended by Technical Editor H. Yu and Senior Editor G. Alici. This work was supported by the National Science Foundation National Robotics Initiative under Grant 2024237 and Grant 1830338. *(Zachary Bons and Gray C. Thomas are Co-first authors.) (Corresponding author: Elliott J. Rouse.)*

Zachary Bons and Elliott J. Rouse are with the Neurobionics Lab, University of Michigan, Ann Arbor, MI 48109-1382 USA (e-mail: zbons@umich.edu; ejrouse@umich.edu).

Gray C. Thomas was with the University of Michigan, Ann Arbor, MI 48109-1382 USA. He is now with the Director of the HERC Lab, Texas A&M University, College Station, TX 77840 USA (e-mail: gray.c.thomas@gmail.com).

Luke Mooney is with the Dephy Inc., Maynard, MA 01754 USA (e-mail: lmooney@dephy.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2023.3334957.

Digital Object Identifier 10.1109/TMECH.2023.3334957

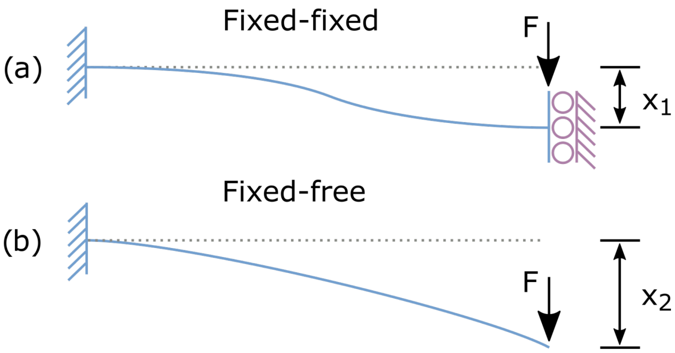


Fig. 1. Changing the loading condition of a beam influences both the stiffness and deflection. A fixed-fixed condition (a) favors high stiffness and low deflection while a fixed-free condition (b) favors low stiffness and high deflection.

(e.g., less than five degrees) and torque limits, which has diminished their success in some applications. While the overall spring performance has been well-predicted in some cases [17], [18], the directionality and limited range of motion are drawbacks of these approaches.

One method to improve the energy density of springs is to modify the loading condition of the elastic elements within the spring. That is, the abovementioned monolith spring designs fully constrain both ends of the spring (i.e., fixed–fixed geometry), which prevents the material from achieving ideal bending, causing lower energy density, and specific energy of the final spring [see Fig. 1(a)]. In addition, they are constrained by the fact that strain rate must be continuous as a function of spring radius, which ultimately limits potential deflection. One way to shift toward more energy-efficient loading conditions and eliminate the strain constraint is to develop springs that can be assembled from multiple parts. For example, Herodotou and Wang [26] proposed a novel design in which one end of the arm is fixed and the other end is hinged. This example is quite efficient (by mass) and the stiffness was accurately predicted by models and finite element analysis (FEA). Drawbacks of these designs are that they add complexity, and have thus far only been developed for extremely high stiffness values (1950 Nm/rad). A fixed-free loading condition trades stiffness for deflection, and thus allows low stiffness at high deflections [see Fig. 1(b)]. Since spring energy is related to the square of deflection, this approach results in greater energy storage when compared to a fixed–fixed beam of the same size.

A spring design that fully exploits an efficient loading condition should, therefore, be capable of achieving low stiffness behaviors while also maintaining a compact form factor and low mass. In addition, a design that employs a convenient mate with transmission components (e.g., fitting inside the timing belt pulley of a belt-drive transmission [27]) would be novel and desirable. Inspired by Mooney et al. [28], we developed a two-element planar spring design that addresses these gaps, is predictable and customizable, and has high energy-per-mass and energy-per-volume efficiency. The preliminary design and findings were presented previously [29], but significant developments will be shared herein.

In this article, we build upon our prior work by the following contributions.

1) Detailing the complete mathematical backing for a compact and lightweight rotary spring design based on simple design indices.
2) Presenting an openly-available design tool for rapidly generating custom springs from simple user inputs [30].
3) Empirically validating the designs of four representative test springs, which more completely span the design space.
4) Demonstrating the effectiveness of both the spring and design tool with respect to the current literature.

These contributions enable rapid adoption of an energy-efficient spring design and lay a foundation for widespread inclusion of springs in lightweight and compact technologies, including but not limited to wearable robots.

II. Spring Design

In this section, we present the design of our spring topology (see Fig. 2) and provide the governing equations that describe the mechanics and geometry of the individual flexures. Our intent is to motivate the rationale for our design decisions as well as provide a guide for other researchers to modify for other applications. In addition, we have developed a spring design tool and graphical user interface (see Section III), which is openly available in [30].

Our spring design [see Fig. 3(a)] comprises a gear-like camshaft in contact with a ring of radial flexure teeth. The teeth protrude inward from a shared outer ring, which is fixed to the spring housing. In operation, the gear-like camshaft rotates relative to the rim ring, imposing a contact force upon the tip of each tooth and causing each of the inward-pointing flexures to bend like a cantilever (i.e., fixed free) beam. The rim of the spring is fixed in place using dowel rods that oppose rotation between the spring and the inner bore of the housing [see Fig. 2(a)]. This is accomplished by designing both the rim ring and the spring housing with complementary semicircular cutouts, and inserting the dowel rods in the hole when the spring and housing are properly aligned. Thus, when the camshaft rotates, it imposes a torque on the spring geometry, which is reacted by the dowel rods, and therefore transferred to the housing (e.g., an output pulley or gear).

A. Camshaft Design

The designed loading condition of this spring enables maximum strain—and thus energy storage—of the bending flexures. Typical monolith springs are limited in this respect due to the constraint that the strain rate must be continuous as a function of spring radius. The relative motion that occurs between the camshaft and spring in our two-part design eliminates this constraint. Taking advantage of this extra degree of freedom, we specifically designed the cam-spring interface to approximate ideal bending: the most efficient loading condition for bending beams.

Ideal bending occurs when a pure moment is imposed along the full length of a beam. This condition can be approximated

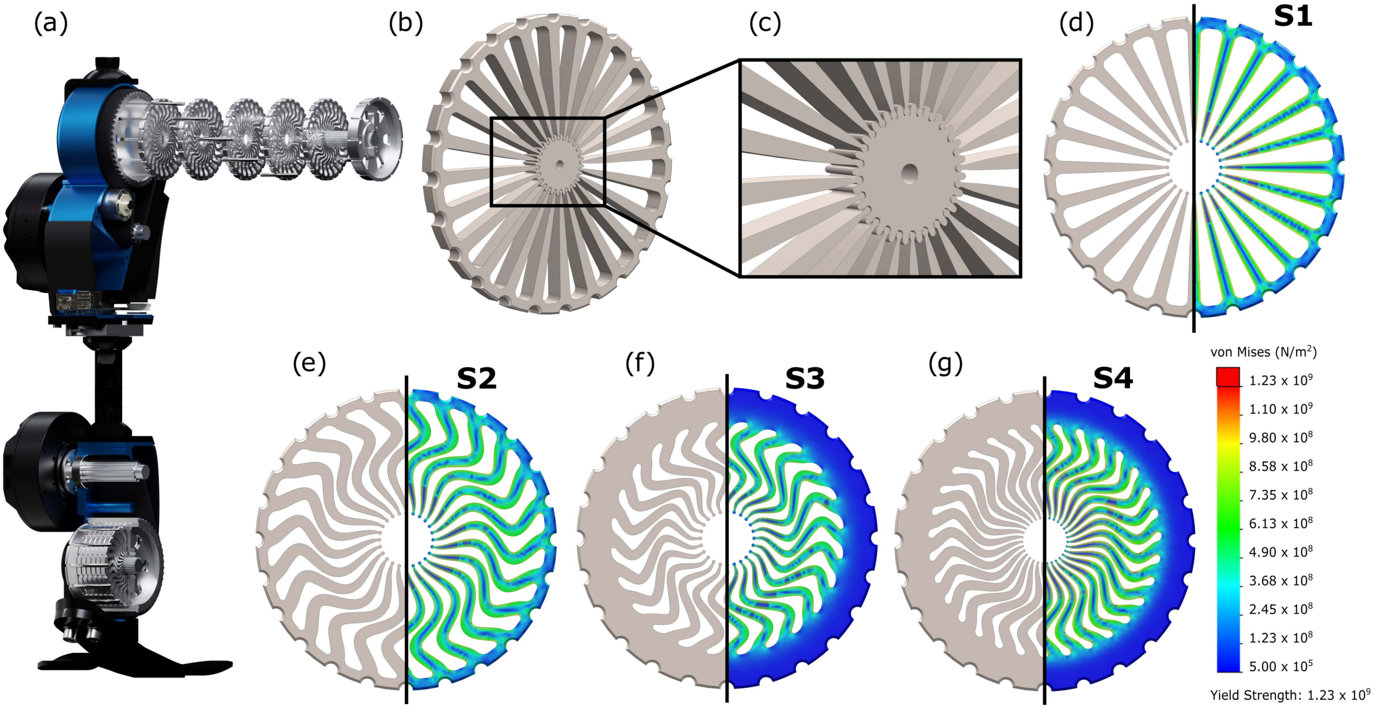


Fig. 2. Rotary spring design based on bending cantilever flexures. (a) Spring neatly mates with transmission components—in this case, a belt pulley in the knee and ankle of the open-source Leg v2. (b) and (c) Flexures mate with a gear-like camshaft, which loads the spring when rotated. d) the original design consists of straight flexures with a tapered profile. (e) and (f) Later designs feature serpentine flexures in addition to the tapered profile. (g) Number of flexures was also optimized for maximal energy storage in the spring.

by applying a force to the free end of the beam, with the force being perpendicular to the neutral axis of the beam. Under this condition, the stress due to the force is small compared with the stress caused by the induced moment, and it therefore resembles a pure moment.

To achieve an applied force that is nominally perpendicular to the length of each flexure, we designed the contact interface to be a circular flexure tip that slides along an involute cam profile. The derivation of the cam profile was accomplished by approximating the path of the flexure tip as parallel to the y-axis [see Fig. 3(b)] throughout the range of spring deflection (i.e., small angle approximation) and iteratively resolved the necessary geometric constraints: at each position along the approximated tip path the camshaft must be tangent to the circular tip and the contact point must be coincident with the line of the tip path. These constraints resulted in a cam profile that is involute to the contact radius and ensures the prescribed perpendicular contact force.

B. Straight Tapered Flexures

To maximize the specific energy of the spring, each inward-pointing flexure is tapered. The tapering law [31] ensures that the entirety of the two bending surfaces [see Fig. 3(b)] reach the desired design stress at peak deflection. Consequently, a significant amount of material can be removed from the standard nontapered beam, increasing the ratio of energy storage to mass. This tapering law governs λ [the distance from the neutral axis to the edge of the beam, see Fig. 3(b)] as a function of x (the distance along each flexure), and is derived from basic beam-bending mechanics [32].

For a generic beam, we know $\sigma = \frac{M\lambda}{I}$, with σ as axial stress, M as applied moment ($F(L-x)$ for our loading condition), λ as above, and I as the second moment of area. In the case of a planar spring, we have $I = \frac{2t\lambda^3}{3}$, with t as the thickness of the spring [see Fig. 3(b)]. To achieve maximum stress along the length of the beam, we choose λ at each cross section such that $\sigma = \sigma_d$, the design stress

$$\sigma_d = \frac{3F(L-x)\lambda}{2t\lambda^3} \tag{1}$$

$$\lambda(x) = \sqrt{\frac{3F(L-x)}{2t\sigma_d}}. \tag{2}$$

This tapering law fully constrains the flexure geometry and can be used to relate spring rate and bending strain energy. By equating bending strain energy and the desired energy storage of the spring, we can then predict deflection properties of the spring.

Assuming that the flexure-cam interface indeed approximates ideal bending, we can easily calculate strain energy. For a generic beam in bending, strain energy (U) is defined by $U = \frac{M\theta}{2}$, with M as previously defined and θ as deflection angle. We also know that $\theta = \kappa L$, where (curvature) $\kappa = \frac{M}{EI}$. Thus, we see that $U = \frac{M^2 L}{2EI}$. For a varying beam profile with small deflections, we

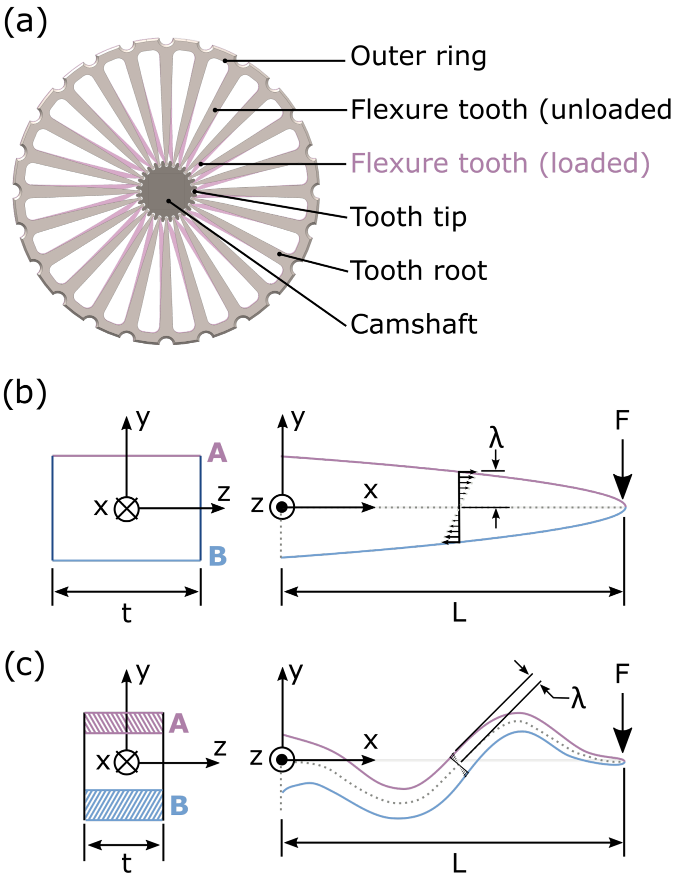


Fig. 3. (a) Spring with labeled parts. (b) Schematic of beam coordinate frames and loading condition for a straight flexure. (c) For a serpentine flexure, where surfaces A and B are the bending surfaces, t is the flexure thickness, L is flexure length, λ is the distance from the neutral axis to the bending surface, and F is the force applied at the tip of the flexure.

can rewrite strain energy as follows:

$$U = \int_0^L \frac{F^2(L-x)^2}{2EI}\,dx. \tag{3}$$

Substituting our definitions of I and λ yields

$$U = \int_0^L \frac{3F^2(L-x)^2}{4Et\sqrt{\frac{3F(L-x)}{2t\sigma_d}}^3}\,dx. \tag{4}$$

Rearranging and simplifying

$$U = \frac{\sigma_d^2 t}{3E}\int_0^L \sqrt{\frac{3F(L-x)}{2t\sigma_d}}\,dx \tag{5}$$

which can be rewritten in closed form as

$$U = \sqrt{\frac{2tFL^3\sigma_d^3}{27E^2}}. \tag{6}$$

Since force (F) is a function of stiffness (k), desired deflection (θ_{des}), the number of flexures (n), and the flexure-camshaft contact radius (r), this can be further simplified to

$$U = \sqrt{\frac{2tk\theta_{\text{des}}L^3\sigma_d^3}{27E^2rn}}. \tag{7}$$

The desired energy storage of a single flexure can be calculated by simply dividing the elastic potential of the full spring by the number of flexures

$$\mathcal{E} = \frac{1}{2n}k\theta_{\text{des}}^2. \tag{8}$$

Therefore, equating the two expressions for energy storage within a flexure—(7) and (8)—and solving for θ_{des} allows us to predict the spring deflection as a function of spring design variables

$$\theta_{\text{des}} = \sqrt[3]{\frac{8tnL^3\sigma_d^3}{27E^2kr}}. \tag{9}$$

Thus, while the tapering law characterizes mass-efficient straight flexures, specific stiffness (k), geometry (r, L, n, t), and material (E, σ_d) constraints directly limit the possible deflection of the spring (9). In addition, the energy density of the spring disks is limited due to the necessary gaps between flexures to accommodate deflection and the gear-like camshaft.

C. Serpentine Tapered Flexures

To maximize energy density while maintaining high specific energy, it is possible to design *serpentine* flexures that follow the tapering law. This results in beams that are effectively longer than the *straight* flexures, yielding higher energy storage through increased flexure deflection for the same outer diameter. A serpentine design also makes better use of the gaps between flexures, resulting in higher overall volume-efficiency in comparison to the straight flexure designs.

To parameterize the design of serpentine flexures, we first rewrite the expression for strain energy (5) by substituting our definition of λ (2)

$$U = \frac{1}{6}\frac{\sigma_d^2}{E}t\left(2\int_0^L \lambda\,dx\right) \tag{10}$$

which can be further generalized to

$$U = \frac{1}{6}\frac{\sigma_d^2}{E}tA. \tag{11}$$

Energy storage is therefore a function of the planar area of the flexures (A), so by appropriately increasing A, we can increase energy storage. Equating desired energy storage (8) with our updated expression for strain energy (11), we easily determine the required planar area (A_{serp}) of a flexure that achieves our desired peak deflection and torque

$$A_{\text{serp}} = \frac{3k\theta_{\text{des}}^2 E}{n\sigma_d^2 t}. \tag{12}$$

Therefore, knowing the required planar area (A_{serp}) and the governing tapered profile, one can define a serpentine flexure that satisfies the constraints. With this approach, spring deflection and stiffness can be prescribed independently, yielding increased design flexibility. To aid in the classification of a spring with such flexures, we introduce two design indices that indicate feasibility of a spring.

1) *Serpentine factor* (f_s): The serpentine factor is calculated as $f_s = \frac{A_{\text{serp}}}{A_{\text{nom}}}$, where A_{nom} is the planar area of the straight

flexures—and describes the sinuosity of a given flexure. If $f_s = 1$, then the desired spring will have straight flexures. If $f_s < 1$, the flexure is undefined, and the diameter of the spring can be decreased while still maintaining the required deflection. Finally, if $f_s > 1$ flexures should have a serpentine shape in order to achieve desired performance.

2) *Density factor* (f_d)*:* A density factor is calculated as $f_d = \frac{A_{\text{serp}} n}{A_{\text{annulus}}}$, where n is the number of flexures in the spring and A_{annulus} is the annular (donut shaped) area, in which the flexure teeth lie. This indicator describes the compactness of the flexures within the spring: a value of 1 would indicate that the spring is a solid disk (all flexures are touching) whereas a value of 0 would represent a spring with no flexure teeth. In the testing performed, we have had difficulty designing springs that do not self-intersect with $f_d > 0.55$.

D. Additional Factors

Several other design parameters have large effects on energy-storage potential, namely, the number of flexures and the flexure-camshaft contact radius. First, increasing the number of flexures within the spring reduces the empty space between flexures—similar to the serpentine concept—and thus increases energy density. Second, decreasing the contact radius makes space for longer flexures, and therefore also increases the energy density of the spring.

The direct relationship between these parameters and spring deflection can be seen by returning to the expression for desired spring deflection (9). Thus, even when all geometry and material parameters are fixed, increasing the number of flexures (n) further increases deflection and therefore energy storage. Similarly, decreasing contact radius (r) directly improves deflection, and it also increases L ($L = R - r$, R being the root radius), which again increases spring deflection. Practical limitations typically govern the possible number of flexures and a feasible contact radius (e.g., nearness constraints of flexure tips). Therefore, a proper selection of these parameters can greatly enhance spring performance.

To summarize, by designing springs with *ideal bending* of *tapered* flexures, we can achieve high specific energy. Selecting the *optimal number* of flexures and imposing a *serpentine* geometry increases the energy density.

III. Flexure Profile Optimization

Since serpentine geometries have many degrees of freedom, we use optimization to automatically define the flexures and offload the burden of geometry selection from the user. Using our software, we can quickly generate one of the many spring profiles that meets desired specifications. Our approach is based on nonlinear optimization that is constrained to achieve the target planar flexure area (among other requirements). We developed this tool to enable quick and simple adoption of this new spring design, and thus lower the barrier to entry for devices with custom elastic elements (files and instructions available [30]). The tool is implemented as a MATLAB application (available as a MATLAB installation file or as an .exe), and generates output files that can be loaded into complete computer aided design packages (e.g., SolidWorks, Dassault Systèmes, Velizy-Villacoublay, and FR) as a 2-D sketch component from which the solid body of the spring can be extruded in a single operation.

The primary objective of our profile optimization is to design a spring flexure that has the desired serpentine factor (f_s) and thickness profile, thereby meeting the performance requirements of the spring. The spring designs are parameterized such that the planar area (and thus the serpentine factor, stiffness, and total stored energy) can be adjusted by the constrained nonlinear optimization without increasing the packaging volume of the spring. Since the serpentine geometry has many degrees of freedom, the optimization is underdefined without secondary objectives. Thus, the optimization includes constraints that align with the assumptions made during the derivation of the design. Specifically, these constraints impose flexure configurations with straight tips, straight roots, laterally balanced geometry, smooth curves, and low likelihood of self-collision, in addition to the planar area requirement.

We parameterized the shape of our flexure using the center curve and thickness profile—both being inputs to the optimization. The center curve is defined using a cubic spline interpolation with end conditions. Splines were chosen for their robust manipulability. The end conditions predefine the desired zero-slope conditions [see xy-plane in Fig. 3(b)] at both the tip and the root of the flexure, while the interior control points allow precise manipulation of the shape between the two ends. During optimization, the number of control points is held constant, as well as the overall packaging format of the spring (e.g., outer radius of the flexure disk and nominal radius of the gear-like camshaft). Points (x_{edge} and y_{edge}) along the boundary of the flexure area are technically defined by moving a distance of λ (2) in the direction perpendicular to the slope (m) of the spline at each point (x_c and y_c) along the center curve

$$\begin{bmatrix} x_{\text{edge}} \\ y_{\text{edge}} \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} \pm \begin{bmatrix} 1 \\ -m \end{bmatrix} \operatorname{sgn}(m) \frac{\lambda(x_c)}{\sqrt{m^2+1}}. \quad (13)$$

Area is then computed numerically using a high-resolution approximation of the flexure curve.

Due to the presence of many local minima in our objective function (which complicates convergence to a global optimum), we include run time as a user-defined input, and restart the optimizer repeatedly from randomized initial conditions until time expires. The optimization is implemented using MATLAB's fmincon, and the computational requirement causes it to take several seconds to solve one set of initial conditions. We tested the implementation on a 4-core 1.50 GHz processor and have found that setting the run time to 30 s allows the tool to find several viable spring designs. The final spring design is then automatically selected from the solutions of all trials.

Several constraints ensure a feasible solution of the nonlinear optimization. These constraints use a 2-D coordinate system with x pointing outward from the center of the spring disk along the nominal center of the flexure.

1) The x-coordinates of the n interior control points are constrained to be in descending order.

2) The corresponding y-coordinates are constrained to the range $[-\frac{L}{2}, 0]$ or $[0, \frac{L}{2}]$ for odd or even points, respectively. This forces the spline to be roughly centered about the x-axis.
3) The flexure face is constrained to have exactly the desired planar area. As shown in the characterization of the spring design, this is necessary to achieve desired spring performance. It is implemented by limiting the difference between the desired and actual serpentine factors ($f_{s,\mathrm{des}}$, f_s) to less than 0.001.
4) The spring is also constrained to be laterally balanced across the x-axis, which ensures that the spring will have similar performance in both loading directions. This is implemented as follows, where n_c is the number of points along the centerline spline and $y_{c,i}$ is the ith y-coordinate along that spline:

$$\left|\sum_{i=1}^{n_c} y_{c,i}\right| < 0.001. \tag{14}$$

The cost function is defined to penalize a combination of the sharpness of curvature and nearness to self-collision between flexures. Eliminating sharp curves is necessary to avoid the nonlinearities associated with stress concentrations. This is achieved by penalizing high curvature along the centerline spline. Curvature (K) is calculated numerically and the sum of the squares of the curvature is included in the cost

$$c_{\mathrm{sharpness}} = \left|\sum_{i=1}^{n_p} K_i^2\right|. \tag{15}$$

Collision of neighboring flexures leads to premature failure of the spring, so the best spring will have a large minimum distance (d_{min}) between neighboring flexures. The minimum distance is calculated by checking the distance between every point along neighboring curves. The minimum is then stored, and its inverse is added to the cost, making small distances expensive

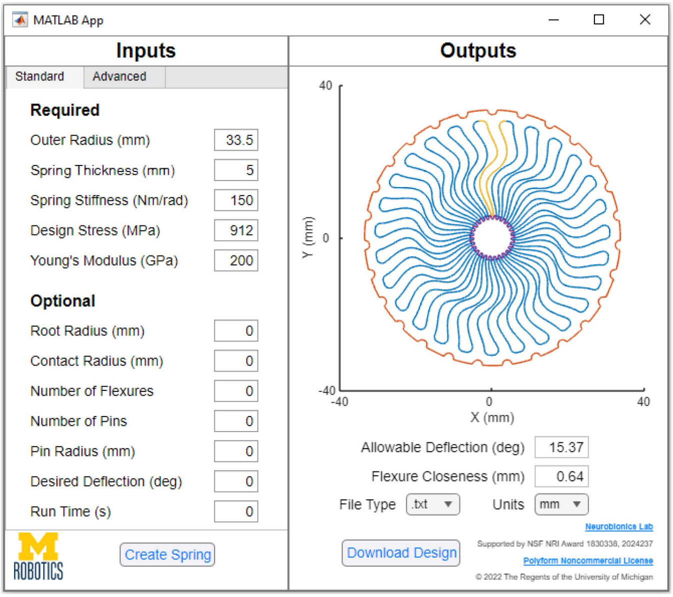
$$c_{\mathrm{nearness}} = \frac{1}{d_{\mathrm{min}}}. \tag{16}$$

The total cost is then calculated as a weighted sum of $c_{\mathrm{sharpness}}$ and c_{nearness}.

To facilitate dissemination of our spring concept, we developed an open-source design tool that implements this profile optimization to automatically design a matching spring and camshaft according to customizable user inputs (see Fig. 4). Inputs include basic geometric constraints, material properties, and performance specifications. The 2-D spring profile is output as Cartesian (i.e., x, y, and z) points in a. txt (or. csv) file, which can be easily imported to CAD software and extruded to achieve the full 3-D model. We utilized this tool to design the springs used in the experimental validation of this spring design.

IV. Hardware Validation of Spring Design Framework

We empirically validated our design framework by comparing four springs with differing geometries and identical desired stiffness coefficients [see Fig. 2(d)–(g)]. The four springs intentionally employ different serpentine factors and root radii (i.e., to simulate springs with a smaller outer ring diameter) to highlight the impact of the flexure geometry on spring stiffness and energy storage capacity. Specifically, the springs tested in this study were as follows (see Table I).



Inputs

Standard | Advanced

Required

Outer Radius (mm): 33.5
Spring Thickness (mm): 5
Spring Stiffness (Nm/rad): 150
Design Stress (MPa): 912
Young's Modulus (GPa): 200

Optional

Root Radius (mm): 0
Contact Radius (mm): 0
Number of Flexures: 0
Number of Pins: 0
Pin Radius (mm): 0
Desired Deflection (deg): 0
Run Time (s): 0

Outputs

Allowable Deflection (deg): 15.37
Flexure Closeness (mm): 0.64
File Type: .txt
Units: mm

Neurobionics Lab
Supported by NSF NRI Award 1830338, 2024237
Polyform Noncommercial License
© 2022 The Regents of the University of Michigan

Fig. 4. Spring design tool for rapid customization according to spring requirements.

1) *Spring One (S1):* It uses 24 straight flexures with a 31 mm root radius, and serves as a baseline comparison for the other designs.
2) *Spring Two (S2):* It uses 24 flexures with a moderate serpentine factor, 1.24, to demonstrate the increased allowable deflection of serpentine flexures when compared to the straight flexures of S1. Its flexures were designed with a 31 mm root radius—equivalent to that of S1.
3) *Spring Three (S3):* It illustrates how serpentine flexures can enable similar performance to the straight flexures of S1 within a smaller enclosed volume. It was also designed with 24 flexures, but with an aggressive serpentine factor of 1.32 and a 26 mm root radius—substantially smaller than that of S1 and S2.
4) *Spring Four (S4):* It was designed to demonstrate the combined effect of using serpentine flexures after optimizing the number of flexures and the flexure-camshaft contact radius for maximum deflection. It has a light serpentine factor of 1.17, 31 flexures, a 5.1 mm contact radius, and the same 26 mm root radius as S3.

All four springs were manufactured from hardened SS 420, and were designed with a target spring rate of 150 Nm/rad, spring thickness of 4.5 mm, and identical outer radius (33.5 mm) to interface with the spring housing.

Although not empirically validated, we also designed our springs to last roughly 100 000 alternating cycles under full load. Using S-N curve estimation as outlined in [33], we determined the design stress (912 MPa) that achieves 100 000 cycles with

TABLE I
DESIGN PARAMETERS AND EXPECTED PROPERTIES OF THE FOUR SPRINGS

Spring Design	Spring Rate (Nm/rad)	Desired Deflection (rad)	Mass (g)	Root Radius (mm)	Contact Radius (mm)	Num. Flex.	Serp. Fact.	Dens. Fact.	Cycles to Failure
S1	150	0.220	57.3	31.0	6.0	24	1.00	0.40	100,000
S2	150	0.253	71.0	31.0	6.0	24	1.24	0.53	100,000
S3	150	0.211	51.8	26.0	6.0	24	1.32	0.53	100,000
S4	150	0.234	60.1	26.0	5.1	31	1.17	0.64	100,000

The mass of all four springs was calculated assuming a 2.5 mm thick rim connecting all flexures on the outer edge.

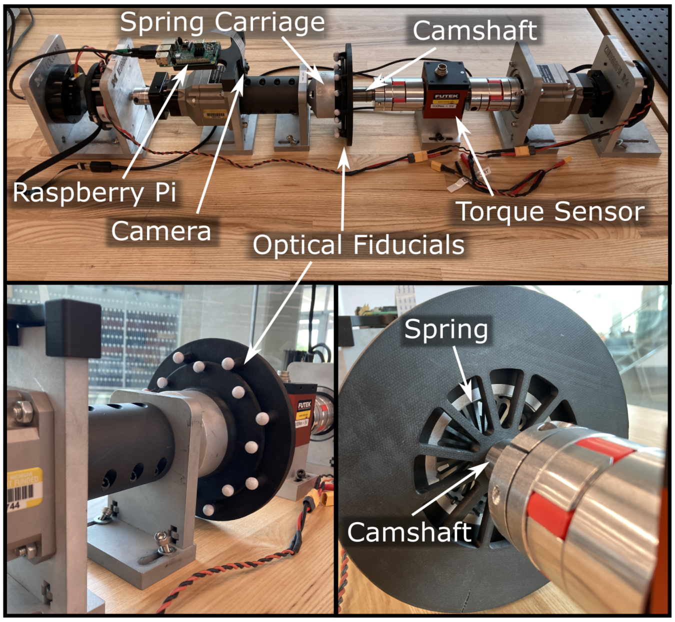


Fig. 5. Testbed used to evaluate spring performance. By tracking optical fiducials with a camera, we were able to measure true deflection of the springs.

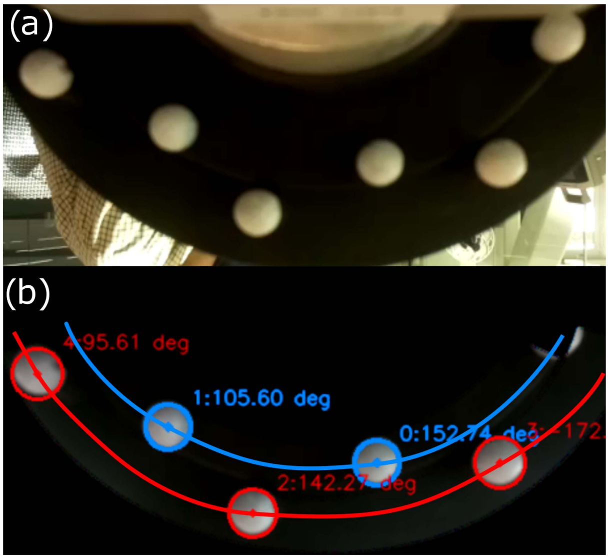


Fig. 6. Optical measurements allowed us to evaluate the true deflection of the springs. (a) High-definition video recorded the movement of the optical fiducials. (b) We tracked each fiducial within the frame, fit the tracked circles to an ellipse, and detected changes in the orientation to calculate angular position.

our material (SS 420). This stress limit was used as an input to the design tool when creating all four springs.

A. Methods

The purpose of our hardware testing was to empirically validate our spring design approach, where each spring was designed with identical stiffness values. Our experimental apparatus (see Fig. 5) deflected the spring by actively using two opposing actuators, while simultaneously measuring the deflection and torque at the spring interface. The springs were held in a housing with complementary semicircular cutouts on the inner bore (see spring design). The applied moments were opposed by inserting dowel rods in the circular cutouts at eight evenly-spaced locations around the spring. Torque was measured with a contactless sensor (TRS605, Futek, Irvine,CA, USA) in series with the spring assembly. The torque sensor output an analog voltage that was sampled by an 16-b analog-to-digital converter at 265 Hz. Rotary motion was provided by two identical brushless dc actuators (ActPack, Dephy Inc, Maynard, MA, USA), each coupled to a transmission (50:1 PL2090-050, Boston Gear, Boston, MA, USA), as used in [29] and [34]. The actuators were controlled in current or position control modes by a microprocessor (RPi 3 Model B+, Raspberry Pi Foundation, Cambridge, U.K.). Both actuators include an encoder; however, deflection within the transmission and testbed setup added error and we therefore obtained the spring deflection separately. Spring deflection was directly measured using a custom optical encoder and image processing algorithm. Two arrays of optical fiducials were used for the measurement: one set was rigidly mounted to the shaft that engaged the spring, while the other set was fixed to the spring housing. Tracking these two fiducial arrays enabled deflection to be sensed locally at the spring interface, improving the quality of our measurements.

A dedicated high-definition camera provided input to our image processing algorithm. The camera (RPi Camera Module 2, Raspberry Pi Foundation, Cambridge, U.K.) recorded the movement of the optical fiducials during motion and we performed an offline analysis to determine relative angular displacement (see Fig. 6). We used filter masks to reduce the image to the areas of interest (the two arcs containing fiducials). Subsequently, we used OpenCV [35] to track the fiducials in the plane of the image, and determined the displacement using best-fit ellipses obtained from a calibration procedure. The accuracy of this approach was validated by comparing the measured displacement to that measured by the motor encoders during unload, and the resulting

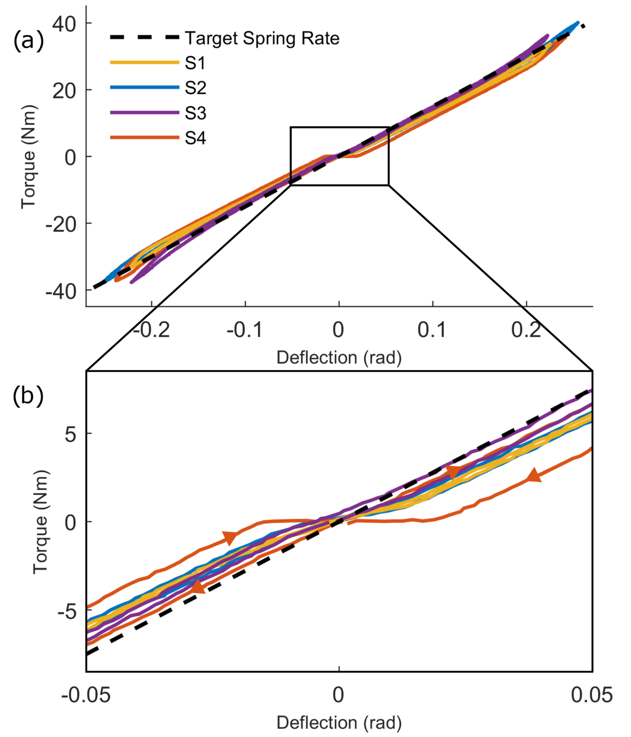


Fig. 7. Measured spring rates of S1, S2, S3, and S4 over (a) each spring's respective operable range and (b) small deflections to show zero-torque crossover behavior. For reference, arrows indicate the loading path for S4, and the target spring rate of 150 Nm/rad is also displayed.

error fell within 95% CI [0.0442, 0.0443] radians. To improve the measurement, we used the relationship between true angle and camera-measured angle (obtained with a no-load calibration procedure) to remove periodic nonlinearities due to optical effects from camera-measured angles. With this additional step, the error of the corrected camera-measured angles reduced to 95% CI [0.00079, 0.00082] radians.

The testing that we performed thoroughly spanned the expected operating ranges of the springs. With each spring, we increased the deflection in one direction to a specified limit, then subsequently decreased the deflection to zero and repeated the process in the opposite direction. We began with two degrees of deflection in both directions and increased the allowable deflection in 1–2° steps until we reached deflections well past the designed limit. Each ramp lasted 5 s, so smaller angles of deflection also had a slower associated angular velocity. By comparing deflection and torque measurements, we quantified the torque–angle relationship of each spring and the maximum ranges of safe operation.

B. Results

The measured performance of the springs closely matched the desired specifications. First, each spring achieves its designed deflection limits without signs of failure (see Fig. 7). In addition, the measured spring rates closely align with the target spring rate (see Table I), falling within 6% of the intended value for all four springs (see Table II). We also quantified energy loss due to hysteresis during the loading and unloading process. At designed deflection (see Fig. 7) the percent energy loss is approximately 5% for the first three springs, but higher for the fourth (S1: 3.77%, S2: 5.36%, S3: 4.82%, and S4: 13.01%). However, at smaller deflections of 0.16 radians, those percentages are significantly lower (S1: 2.30%, S2: 2.40%, S3: 0.84%, and S4: 2.09%).

TABLE II
Spring Rates of the Four Spring Designs During Loading and Unloading in Both Positive and Negative Torque Regimes

Spring Design	Spring Rate (Nm/rad)					Error (%)
	Loading		Unloading		Avg.	
	Pos.	Neg.	Pos.	Neg.		
S1	149.2	150.9	139.4	139.9	144.9	3.43
S2	158.0	151.4	144.1	142.4	149.0	0.68
S3	156.6	171.5	152.1	159.9	160.0	6.68
S4	147.7	155.8	145.2	147.4	149.0	0.65

Average spring rate and percent error are also reported. The desired spring rate was 150 Nm/rad for all four designs.

V. Discussion

A. Contributions

In this article, we introduced a novel two-part spring architecture, along with an open-source design tool that enables rapid customization and implementation of the spring. In addition, we presented an empirical validation of four representative springs to demonstrate the correspondence between our mechanical model and the physical realizations. The intent of our spring design was to fit within the volume of common transmission components used in the open-source Leg [27] and other robotic systems, thus, enabling series elasticity of the joints without increasing their size. This design objective led to a two-part spring architecture comprised of a ring of radially-spaced cantilever flexures in contact with an involute camshaft that lies in the center of the ring.

We made several intentional decisions in the design process to maximize energy storage and density of the spring. First, the two-part design allows relative motion between the camshaft and the spring, which removes the constraint of a continuous strain rate as a function of spring radius. Second, a tapered profile induces equivalent stress along the full length of the flexures, thus eliminating unnecessary mass. Third, serpentine flexures allow greater deflection and length—and thus energy storage—when compared with straight flexures of the same spring radius. To our knowledge, these innovations result in a new class of torsion springs with the highest specific energy and energy density to date[1,2] (see Fig. 8).

[1] Traditional, wire-type torsion springs (e.g., clothespin springs) have specific energy and energy density of around 40 $\frac{J}{kg}$ and 0.15 $\frac{J}{cm^3}$, respectively, but we did not include them in the comparison because they are typically unidirectional and we were unable to find reliable data in the stiffness regimes that are common in robotic systems.

[2] For reference, linear/prismatic diesprings can have specific energy over 200 $\frac{J}{kg}$ and energy density greater than 0.3 $\frac{J}{cm^3}$, but the additional fixturing elements required to convert linear motion to rotary motion drastically decrease these values [13].

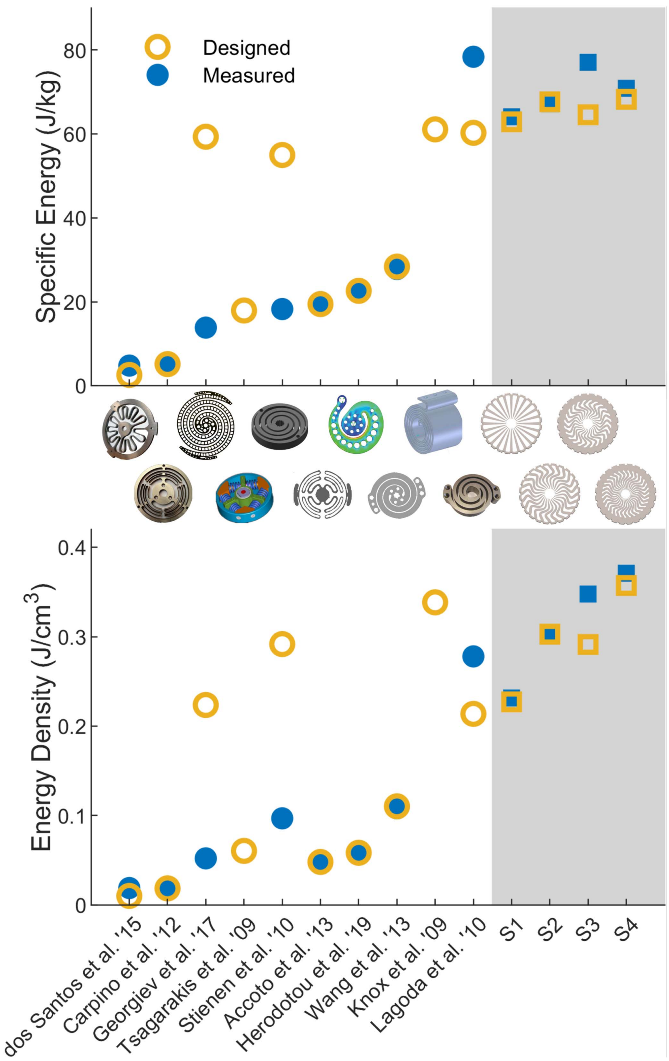


Fig. 8. Performance of cantilever-beam rotary springs (gray region) by energy storage per unit mass and per unit volume in the context of the current literature. It should be noted that the mass and volume of the S3 and S4 springs were estimated for an outer rim thickness comparable with that of S1 and S2 rather than using the unnecessarily large rim than was dictated by our testing apparatus.

We achieved high agreement between our mechanical model and our empirical results. The mechanical model of the spring mechanics involved several simplifying assumptions, including the forces being perpendicular to the nominal flexure, ideal bending along the flexure, and small angles of deflection. The low error between the empirically-determined and desired stiffness coefficients (1–6%) indicates these assumptions had minimal effect in the torsion spring mechanics. Furthermore, the springs achieved their respective desired deflections and the observed backlash was minimal despite the two-part design [see Fig. 7(a) and (b)]. We also accounted for fatigue life by appropriately modifying the design stress, according to an estimated S-N curve [33]. All four springs were designed to last over 100 000 loading cycles.

To facilitate the implementation of springs within mechanical and robotic systems, we developed an open-source design tool that outputs custom spring designs—according to the user's inputs—in a matter of minutes [30]. The tool couples an optimization routine with our mechanical model to automatically design a spring and camshaft that satisfy the design requirements. The spring geometry can then be exported for direct implementation in solid modeling software for design. Future researchers that wish to include our springs in their designs can now do so without the overhead of mathematical derivations or iterative FEA. While our empirical validation was limited to springs with stiffnesses in the neighborhood of 150 Nm/rad, we have used the design tool to gain insight about viable stiffness ranges for this architecture. On the low end, springs with stiffness values of 30 Nm/rad appear to be practical, but below that value the geometry becomes less compact and machining tolerances have an increasing effect on spring properties. On the upper end, the range of achievable stiffnesses is constrained by three factors: material selection, housing diameter, and spring thickness. If the spring application is unconstrained, arbitrarily high stiffness values could be achieved.

B. Broader Applicability

This spring design is highly customizable and easily implemented, making it suitable for a broad range of applications. Immediate areas of impact will likely include series and parallel elastic actuators, and other mechanisms in which torsional compliance is desirable. The unique combination of compactness and high-energy storage make this spring particularly useful in systems that prioritize low mass and volume (e.g., wearable robotic systems, including exoskeletons and prostheses), but it is also relevant in serial-link manipulators, humanoids, and other mobile robots where mass and volume are design-driving factors. As an example of the potential impact of our approach, we compared our spring architecture with five existing spring designs used in powered mobile robots (see Fig. 9). We used our open-source tool to design springs that matched the stiffness and deflection requirements of the springs in a lower limb rehabilitation exoskeleton [16], an upper limb rehabilitation exoskeleton [15], a lower limb robotic orthosis [5], a planar bipedal robot [4], and the humanoid Robonaut 2 [37]. In every case, the output of our design tool resulted in a spring that was both lighter and more compact than the original spring (see Table III). For each spring used in this comparison, we searched for information regarding the fatigue life of the original design to match the number of cycles in our comparison; however, we were only able to find the relevant details for one spring [5] and in this case we matched their prescribed number of roughly 42 000 cycles. Notably, the process of designing these spring examples with our open-source design tool required less than 10 min from knowing the design requirements to completing the solid model design.

Further modifications to the spring design could increase applicability. For example, implementing series connections of springs would yield even greater deflections [20], or capitalizing on the underconstrained interface between the cam and the spring could result in a sliding clutch or variable stiffness

TABLE III
IMPLEMENTATION OF OUR SPRING DESIGN TOOL WITH VARIOUS TYPES OF ROBOTS

Application	Ref.	Spring	Spring Rate (Nm/rad)	Defl. (rad)	Mass (g)	OD (mm)	Thick. (mm)	Improvement (%) Mass	Vol.
Lower-limb rehab exo	[16]	Original	353	0.283	235	75	15	12	42
		New Design			206	70	10		
Upper-limb rehab exo	[15]	Original	150	0.332	150	60	10	14	10
		New Design			129	60	9		
Lower-limb robotic orthosis	[5]	Original	250	0.240	370	90	23.5	78	90
		New Design			79.9	76	3.2		
Planar Bipedal Robot	[4]	Original	150	0.262	150	67	25	35	78
		New Design			97.5	50	10		
Humanoid Robot	[37]	Original	344	0.087	45.1	33.6	3.1	53	65
		New Design			21.2	20	3.1		

For each of the listed applications, we used the open-source design tool to design a spring that matched the stiffness and deflection specifications of the robot, and compared the size and mass of our design to that of the original spring. Springs are displayed according to the scale shown at the bottom right.

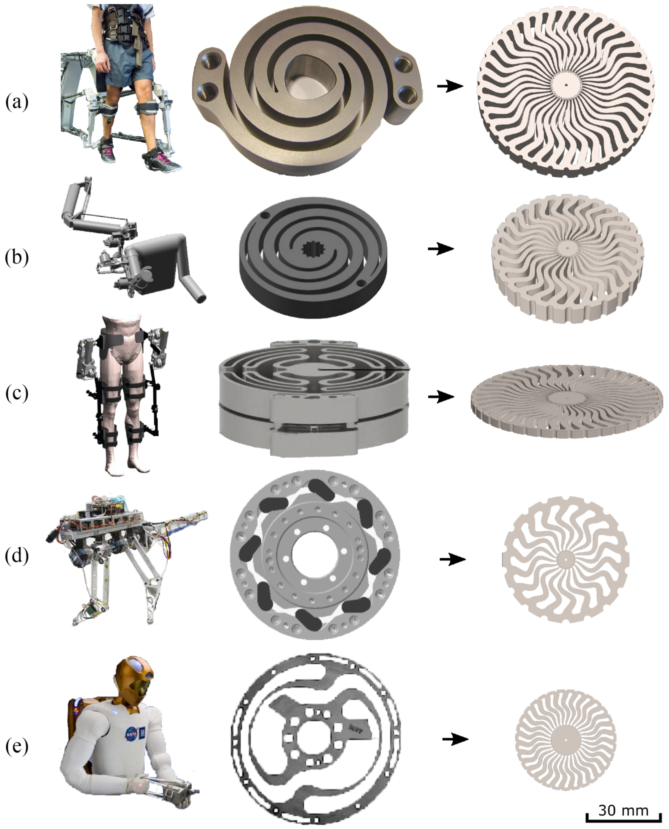


Fig. 9. Effect of our novel spring architecture on wearable robots. In each row, the application is shown, followed by the original spring for that design and a spring designed by our open-source design tool with equivalent performance specifications. All springs are represented according to the scale at the bottom right. (a) LOPES II exoskeleton [36] uses SEA [16]. (b) Limpact upper-limb rehabilitation exoskeleton uses a spring as part of a rotational hydroelastic actuator [15]. (c) Lower limb robotic orthosis [5] uses SEAs at the joints. (d) Planar bipedal robot [4] uses rubber Neidhart springs. (e) Robonaut 2 humanoid [37] employs series elastic arms.

mechanism. In addition, the bending beam paradigm could be directly applied to prismatic springs, which could be useful as a high-efficiency alternative to springs in series with linear actuators, or in mechanisms, such as suspension systems.

C. Limitations

Our design framework produces compact springs that match user specifications, but future work could improve or further validate the approach. First, the sources of hysteresis could be investigated in depth. It is likely that the hysteresis is due in large part to the sliding contact between the camshaft and the spring; however, the friction could be significantly reduced by lubricating the contact surfaces or improving their surface finish. Interestingly, the fourth test spring (S4) had much larger energy loss due to hysteresis than the other three springs (13% compared with ∼5%). This behavior is clearly displayed by the measured torque-deflection curve for S4, where the collinearity of the loading paths confirms that the difference is due to hysteretic loss, and not a confounding factor, such as backlash [see Fig. 7(b)]. The major differences in the design of this spring include a larger number of flexures (30% increase), a smaller contact radius (15% decrease), and a smaller flexure-tip diameter (33% decrease). Thus, one plausible explanation for the difference in hysteresis could stem from the increased friction due to a larger number of contact surfaces.

Second, the slope of the measured torque-deflection curves is not perfectly constant, and it generally increases as the springs approach their deflection limits [see Fig. 7(a)]. While the spring rate is likely sufficiently linear for the majority of spring applications, there are certain situations in which high precision may be needed. For such situations, the linearity could potentially be improved by modifying the shape of the flexure tip and/or the camshaft.

VI. CONCLUSION

In this article, we presented a novel spring design characterized by a ring of radially-extending tapered cantilever beams in contact with a gear-like camshaft. This new paradigm yields the most energy-efficient rotary springs to date. In addition, we introduced a design tool that will facilitate rapid adoption and customization of this technology. We also tested four representative springs on a custom testbed to demonstrate the high fidelity of our design framework. This work increases the potential for incorporating springs into compact mechanical systems.

References

[1] J. Pratt, B. Krupp, and C. Morse, "Series elastic actuators for high fidelity force control," *Ind. Robot*, vol. 29, no. 3, pp. 234–241, 2002.

[2] C. Fitzgerald, "Developing baxter," in *Proc. Conf. Technol. Practical Robot Appl.*, 2013, pp. 1–6.

[3] K. Kong, J. Bae, and M. Tomizuka, "Control of rotary series elastic actuator for ideal force-mode actuation in human–robot interaction applications," *IEEE/ASME Trans. Mechatron.*, vol. 14, no. 1, pp. 105–118, Feb. 2009.

[4] T. Kamioka, H. Shin, R. Yamaguchi, and M. Muromachi, "Development and analysis of a biped robot with prismatic compliance," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 10398–10404.

[5] D. Accoto, G. Carpino, F. Sergi, N. L. Tagliamonte, L. Zollo, and E. Guglielmelli, "Design and characterization of a novel high-power series elastic actuator for a lower limb robotic orthosis," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 10, 2013, Art. no. 359. [Online]. Available: https://doi.org/10.5772/56927

[6] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Hum. Robot Interact. Cooperative Robots*, 1995, pp. 399–406.

[7] M. M. Williamson, "Robot arm control exploiting natural dynamics," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1999.

[8] G. A. Pratt, "Low impedance walking robots," *Integrative Comp. Biol.*, vol. 42, no. 1, pp. 174–181, 2002.

[9] M. K. Shepherd and E. J. Rouse, "The VSPA foot: A Quasi-passive ankle-foot prosthesis with continuously variable stiffness," *Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 12, pp. 2375–2386, Dec. 2017.

[10] C. Hubicki et al., "ATRIAS: Design and validation of a tether-free 3D-capable spring-mass bipedal robot," *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1497–1521, 2016.

[11] F. Negrello et al., "Design and characterization of a novel high-compliance spring for robots with soft joints," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics*, 2017, pp. 271–278.

[12] S. Yoon, S. Kang, S. Kim, Y. Kim, M. Kim, and C. Kim, "Safe arm with MR-based passive compliant joints and visco-elastic covering for service robot applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 2191–2196.

[13] N. G. Tsagarakis, M. Laffranchi, B. Vanderborght, and D. G. Caldwell, "A compact soft actuator unit for small scale human friendly robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 4356–4362.

[14] B. T. Knox and J. P. Schmiedeler, "A unidirectional series-elastic actuator design using a spiral torsion spring," *J. Mech. Des.*, vol. 131, no. 12, 2009, Art. no. 125001, doi: 10.1115/1.4000252.

[15] A. H. Stienen, E. E. Hekman, H. ter Braak, A. M. Aalsma, F. C. van der Helm, and H. van der Kooij, "Design of a rotational hydroelastic actuator for a powered exoskeleton for upper limb rehabilitation," *Trans. Biomed. Eng.*, vol. 57, no. 3, pp. 728–735, 2010.

[16] C. Lagoda, A. C. Schouten, A. H. Stienen, E. E. Hekman, and H. van der Kooij, "Design of an electric series elastic actuated joint for robotic gait rehabilitation training," in *Proc. IEEE 3rd RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, 2010, pp. 21–26.

[17] S. Wang, C. Meijneke, and H. van der Kooij, "Modeling, design, and optimization of mindwalker series elastic joint," in *Proc. IEEE 13th Int. Conf. Rehabil. Robot.*, 2013, pp. 1–8.

[18] N. Georgiev and J. Burdick, "Design and analysis of planar rotary springs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4777–4784.

[19] S. Yoo, J. Lee, J. Choi, G. Chung, and W. K. Chung, "Development of rotary hydro-elastic actuator with robust internal-loop-compensator-based torque control and cross-parallel connection spring," *Mechatronics*, vol. 43, pp. 112–123, 2017.

[20] G. Carpino, D. Accoto, F. Sergi, N. Luigi Tagliamonte, and E. Guglielmelli, "A novel compact torsional spring for series elastic actuators for assistive wearable robots," *J. Mech. Des.*, vol. 134, no. 12, 2012, Art. no. 121002, doi: 10.1115/1.4007695.

[21] N. Paine et al., "Actuator control for the NASA-JSC Valkyrie humanoid robot: A decoupled dynamics approach for torque control of series elastic robots," *J. Field Robot.*, vol. 32, no. 3, pp. 378–396, 2015.

[22] W. M. dos Santos, G. A. Caurin, and A. A. Siqueira, "Design and control of an active knee orthosis driven by a rotary series elastic actuator," *Control Eng. Pract.*, vol. 58, pp. 307–318, 2017.

[23] T. Kim, K. Shi, and K. Kong, "A compact transmitted-force-sensing series elastic actuator with optimized planar torsional spring for exoskeletons Taeyeon," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2021, pp. 572–577.

[24] G. G. Fiorezi, J. dos Santos de Moraes, P. H. Fabriz Ulhoa, and R. M. de Andrade, "Biomimetic design of a planar torsional spring to an active knee prosthesis actuator using FEM analysis," in *Proc. 1st Int. Electron. Conf. Actuator Technol.*, 2020, pp. 1–11.

[25] J. P. Cummings, D. Ruiken, E. L. Wilkinson, M. W. Lanighan, R. A. Grupen, and F. C. S. IV, "A compact, modular series elastic actuator," *J. Mechanisms Robot.*, vol. 8, pp. 1–11, 2016.

[26] P. Herodotou and S. Wang, "Design, modelling, and experimental evaluation of a compact elastic actuator for a gait assisting exoskeleton," in *Proc. IEEE 16th Int. Conf. Rehabil. Robot.*, 2019, pp. 331–336.

[27] A. F. Azocar, L. M. Mooney, J.-F. Duval, A. M. Simon, L. J. Hargrove, and E. J. Rouse, "Design and clinical implementation of an open-source bionic leg," *Nature Biomed. Eng.*, vol. 4, no. 10, pp. 941–953, 2020.

[28] L. Mooney, K. A. Pasch, and T. D. Doan, "Planar torsion spring for knee prostheses and exoskeletons," Aug. 2017, U.S. Patent App. 15/402,186.

[29] Z. Bons, G. C. Thomas, L. M. Mooney, and E. J. Rouse, "A compact, two-part torsion spring architecture," in *Proc. Conf. Robot. Autom.*, 2023, pp. 7461–7467.

[30] The spring design tool and associated source code. May 12, 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10261479

[31] M. K. Shepherd and E. J. Rouse, "Design and validation of a torque-controllable knee exoskeleton for sit-to-stand assistance," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 4, pp. 1695–1704, Aug. 2017.

[32] B. J. Goodno and J. M. Gere, *Mechanics of Materials*, 9th Ed., Boston, MA, USA: Cengage Learn., 2013.

[33] J. A. Collins, H. Busby, and G. Staab, *Mech. Des. of Mach. Elements and Machines*, 2nd Ed., Hoboken, NJ, USA: Wiley, 2010.

[34] E. A. Bolívar-Nieto, G. C. Thomas, E. Rouse, and R. D. Gregg, "Convex optimization for spring design in series elastic actuators: From theory to practice," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 9327–9332.

[35] G. Bradski, "The OpenCV library," *Dr Dobb's J. Softw. Tools*, vol. 25, pp. 120–123, 2000.

[36] G. Zhao, M. Ahmad Sharbafi, M. Vlutters, E. van Asseldonk, and A. Seyfarth, "Bio-inspired balance control assistance can reduce metabolic energy consumption in human walking," *Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 9, pp. 1760–1769, 2019.

[37] M. A. Diftler et al., "Robonaut 2 - The first humanoid robot in space," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2178–2183.

Zachary Bons (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from Brigham Young University, Provo, UT, USA, in 2021. He is currently working toward the Ph.D. degree in mechanical engineering awith the University of Michigan, Ann Arbor, MI, USA.

His research interests include the design, development, and control of robotic exoskeletons and prostheses with the purpose of improving mobility and quality of life.

Mr. Bons is a Member of the Neurobionics Lab.

Gray C. Thomas (Member, IEEE) received the B.S. degree in engineering: robotics from Olin College, Needham, MA, USA, in 2012, and the Ph.D. degree in mechanical engineering from the University of Texas at Austin, Austin, TX, USA, in 2019.

He was with Robotics Department, University of Michigan, Ann, Arbor, MI, USA. He is currently an Assistant Professor with the J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA, where he directs Human Empowering Robotics and Control Lab.

Luke Mooney received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2012, 2014, and 2016, respectively.

He was a Doctoral Researcher with Biomechatronics Group, MIT Media Lab, Cambridge, MA, USA. His research interests include wearable robotics and actuator design.

Dr. Mooney is the Co-Founder of Dephy, Inc., Boxborough, MA, USA, a robotics design firm based out of Maynard.

Elliott J. Rouse (Senior Member, IEEE) received the B.S. degree in mechanical engineering from The Ohio State University, Columbus, OH, USA, in 2007, and the M.S. and Ph.D. degrees in biomedical engineering from Northwestern University, Evanston, IL, USA, in 2009 and 2012, respectively.

He joined the Massachusetts Institute of Technology, Cambridge MA, USA, where he was a Postdoctoral Fellow till 2014. He is currently an Associate Professor with the Departments of Robotics and Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA. He was a Faculty with Shirley Ryan AbilityLab, Northwestern University, Chicago, IL, USA. His research interests include precision machine design, control, biomechanical modeling, psychophysics, and the use of these tools to develop meaningful exoskeletons and robotic prostheses that advance mobility and quality of life.