This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929984, IEEE Robotics and Automation Letters

Computational Design of Statically Balanced Planar Spring Mechanisms

Takuto Takahashi[1,3], Jonas Zehnder[3], Hiroshi G. Okuno[1], Shigeki Sugano[1], Stelian Coros[2], and Bernhard Thomaszewski[3]

***Abstract*—Statically balanced spring mechanisms are used in many applications that support our daily lives. However, creating new designs is a challenging problem since the designer has to simultaneously determine the right number of springs, their connectivity, attachment points, and other parameters. We propose a novel optimization-driven approach for designing statically balanced mechanisms in an interactive, semi-automatic way. In particular, we describe an efficient method for design optimization based on the principle of constant potential energy, an automated sparsification method for spring topology design, and a null-space exploration scheme that enables user to navigate the local space of design alternatives. We demonstrate our design system on a set of simulation examples and several manufactured prototypes.**

***Index Terms*—Mechanism Design, Product Design, Development and Prototyping,**

I. INTRODUCTION

PIXAR'S short film "Luxo Jr." is a landmark of computer animation. The story is centered around a desk lamp brought to life through expressive animation that draws on its ability to effortlessly transition between, and balance in, arbitrary poses [1]. The real-world inspiration for Luxo Jr. is the well-known Anglepoise lamp [2], which is statically balanced in any mechanically admissible pose despite the significant load induced by the lamp's head. Similar to the Anglepoise lamp, many kinds of balanced mechanisms use counterweights or springs for maintaining static equilibrium throughout a wide range of configurations.

Statically balanced mechanisms are employed in many applications that support our daily lives. In appliances, spring-loaded mechanisms allow us to effortlessly manipulate heavy doors; in robotics, gravity-compensation mechanisms facilitate the motion of robotic arms, thus allowing for smaller, lighter, and less expensive actuators; and in medical engineering, passively balanced orthoses can help patients to stand and walk. The common trait among all these applications is energy efficiency—statically balanced mechanisms reduce the work required to achieve a desired function.

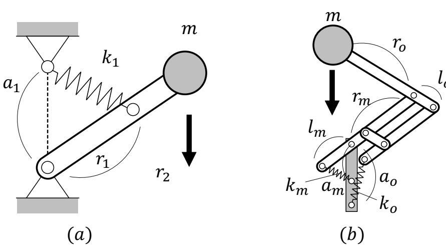


Fig. 1. Templates of statically balanced spring mechanisms.
(a) A basic gravity equilibrator with one spring. (b) An Anglepoise mechanism with two springs.

Two main approaches exist for designing statically balanced mechanisms: *using counterweights* or using *elastic elements* such as springs [3]. Using counterweights is comparatively simple to implement; however, counterweights increase the total inertia of the mechanism, which may degrade the dynamic performance of the mechanism and its ability to withstand external forces.

In this work, we focus on using elastic elements to achieve static balance. One well-known example in this class is the basic gravity equilibrator [4] shown in Fig. 1(a). The static balance condition for this single-spring mechanism is given as

$$mgr_2 = r_1 k_1 a_1 \ . \tag{1}$$

Similarly, the conditions for the double-spring Anglepoise mechanism [2] shown in Fig. 1(b) are

$$\begin{aligned} mgr_m &= l_m k_m a_m \ , \\ mgr_o &= l_o k_o a_o \ . \end{aligned} \tag{2}$$

The equilibrium conditions for the above examples can be derived using simple geometric reasoning. However, there are only few mechanisms for which the static balance conditions are explicitly known.

Instead of using explicit balance conditions, an alternative approach is to require that the total potential energy must be constant across all admissible configurations of the mechanism [4], [5]. Several works have proposed design methods based on this principle, e.g., to determine the stiffness of torsional springs in a graphical manner [6]. As an alternative to rigidly-articulated mechanisms, previous works have investigated the

Manuscript received: February, 24, 2019; Revised May, 31, 2019; Accepted July, 15, 2019.

This paper was recommended for publication by Editor Paolo Rocco. Name upon evaluation of the Associate Editor and Reviewers' comments.

Takuto Takahashi is supported by the Graduate Program for Embodiment Informatics funded by JSPS at Waseda University. Hiroshi G. Okuno is supported by Kakenhi 19H00750. Bernhard Thomaszewski is supported by the NSERC Discovery Grants and Discovery Accelerator Awards program. Jonas Zehnder is supported by a Google Excellence scholarship.

[1]Department of Modern Mechanical Engineering, Waseda University, Japan. chobby75@akane.waseda.jp, okuno@nue.org, sugano@waseda.jp

[2]Department of Computer Science, ETH Zurich, Switzerland. scoros@gmail.com

[3]Department of Computer Science and Operations Research, Université de Montréal, Canada. jonas.zehnder@umontreal.ca, bernhard@iro.umontreal.ca

Digital Object Identifier (DOI): see top of this page.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929984, IEEE Robotics and Automation Letters

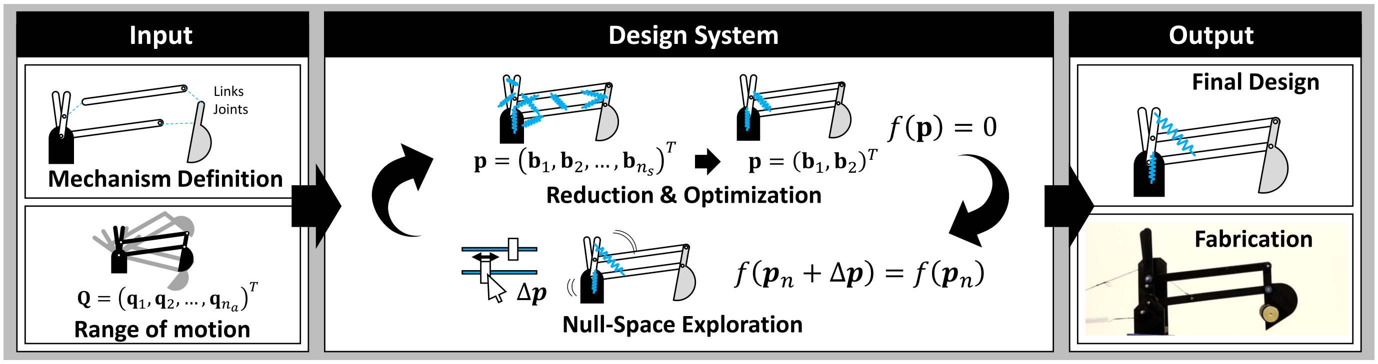


Fig. 2. Overview of the proposed design system. *Left*: Our method accepts as input a rigidly-articulated mechanism (*top*) and a set of configurations (*bottom*) outlining the desired range of motion. *Middle*: We automatically suggest candidate springs and optimize their parameters to achieve static balance. By navigating the null-space of our static balance objective, the user can interactively explore different design alternatives. *Right*: final design (*top*) and physical prototype (*bottom*).

shape of the links in compliant mechanisms to achieve static balance; see, e.g., [7]. One limitation of many existing design methods for statically balanced mechanisms is, however, that they assume the *topology* of the mechanism—the number of elastic elements and how they connect between links—to be known in advance.

Our goal is to establish a computational approach for discovering new statically balanced spring mechanisms. Based on the constant energy principle, we develop a fast, optimization-driven method that enables interactive exploration of spring topology and parameters. In order to achieve this goal, our method relies on the following technical contributions:

- A formulation for automatic optimal design of statically balanced spring mechanisms within a user-defined space of configurations.
- A topology simplification method based on a sparse regularizer for reducing the number of required springs.
- A *null-space exploration* method for navigating the space of first-order feasible design variations.

II. RELATED WORK

The design of statically balanced spring mechanisms is a fundamental task in many fields. In robotics, for example, gravity compensation mechanisms play a critical role for efficiency [3], [8], [9], [10]; the development of balanced exoskeletons is another challenging research topic [11], [12]. Furthermore, combining the concepts of static balance and reconfigurability [13], [14] can open the door to many new applications.

Fueled by the increasing availability of additive manufacturing equipment, there has recently been a growing research interest in optimization-driven tools that simplify the design of rigidly-articulated mechanisms [15], [16], [17], compliant mechanisms [18], and cable-driven mechanisms [19] with desired output motion. Our work falls into this category of interactive design tools that, instead of having the user specify parameters manually, take an optimization-driven approach guided by high-level objectives in order to simplify the design task.

Beyond pure kinematics, designing mechanisms that are optimized with respect to peak torques, joint forces, and actuator requirements is a topic of ongoing research; see, e.g., [20], [21], [22], [23], [24] and references therein. To simplify the design of statically balanced compliant mechanisms, several works have focused on conventional mechanisms, augmenting pin joints with torsional springs [5], [6], [25], [26]. Our approach differs from these works in that it uses linear springs whose free lengths and attachment points are optimized in an interactive, user-guided manner.

III. METHOD

In this work, we propose a computational approach for designing statically balanced mechanisms in a semi-automated way; see Fig. 2 for an overview of our system.

Our method accepts as input a conventional mechanism consisting of rigid links and joints. The user also defines the desired range of static balance for the mechanism through a set of postures $\mathbf{Q}$. During interactive design, the user explores the space of feasible, i.e., statically balanced, designs using high-level commands and objectives in the context of a graphical user interface (GUI). For example, the user can choose to insert springs between specific links or receive suggestions from our system on which springs to remove. Our system then interactively computes optimized design parameters $\mathbf{p}$ that approximate the user-defined objectives as closely as possible while satisfying the static balance constraints.

A. Mechanical Model

We model planar mechanisms using rigid links, elastic springs, and mechanical joints. A mechanism consists of n_c links, each of which is a rigid body with three degrees of freedom

$$\mathbf{a}_i = \left(\theta, \mathbf{z}^T\right)^T , \tag{3}$$

where θ is an angle and $\mathbf{z}$ is the 2D center position of the body in global coordinates. For notational convenience, we concatenate the variables for all n_c links into one vector

$$\mathbf{q} = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{n_c})^T . \tag{4}$$

We model elastic elements as linear Hookean springs defined by six parameters

$$\mathbf{b}_i = (l, k, \hat{\mathbf{x}}_A, \hat{\mathbf{x}}_B)^T , \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929984, IEEE Robotics and Automation Letters

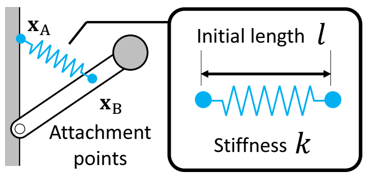


Fig. 3. The parameters of a spring are its initial length l, stiffness k, and attachment locations $\hat{\mathbf{x}}_A$ and $\hat{\mathbf{x}}_B$; the parameters of all springs are concatenated into the design parameter vector $\mathbf{p}$.

where k is the spring stiffness, l its initial length, and $\hat{\mathbf{x}}_A$ and $\hat{\mathbf{x}}_B$ are attachment locations in local coordinates on the two links, A and B, that the spring connects; see Fig. 3. We likewise concatenate the parameters of all n_s springs into a single parameter vector

$$\mathbf{p} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n_s})^T \ . \tag{6}$$

Joints are modeled as constraints on the geometric relation between a pair of links. We restrict considerations to pin joints and sliding joints in this work and define corresponding constraint functions as

$$c_{\text{pin}}(\mathbf{q}) = \mathbf{x}_A(\mathbf{q}) - \mathbf{x}_B(\mathbf{q}),$$
$$c_{\text{slider}}(\mathbf{q}) = (\mathbf{x}_A(\mathbf{q}) - \mathbf{x}_B(\mathbf{q}))^T \mathbf{v}(\mathbf{q}) \ ,$$

where $\mathbf{x}_A(\mathbf{q})$ is the position of the joint on link A in world space (and analogously for $\mathbf{x}_B$), and $\mathbf{v}(\mathbf{q})$ is the world-space version of the unit vector $\mathbf{v} = (-\sin\phi_A, \cos\phi_A)^T$, indicating the sliding direction on link A through a local angle ϕ_A. We collect the constraint functions of the n_j joints into a global vector $\mathbf{C} = (c_1, c_2, \ldots, c_{n_j})^T$. In order to eliminate global rigid-body motion, we introduce additional constraints to fix the degrees of freedom of individual links.

Given a fully-constrained mechanism, we find the configuration $\mathbf{q}$ that satisfies all constraints by solving the unconstrained minimization problem

$$\min_{\mathbf{q}} \quad \frac{1}{2}\mathbf{C}(\mathbf{q})^T \mathbf{C}(\mathbf{q}) \tag{7}$$

using Newton's method.

B. Target Space

The target space is the set of configurations in which the mechanism should be statically balanced. In the discrete setting, we represent this space using n_a individual target configurations $\mathbf{q}_i$ that we stack as $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{n_a})^T$. To facilitate the creation of $\mathbf{Q}$, we allow the user to specify target angles for one or several joints per target configuration and compute the remainder of $\mathbf{q}_i$ automatically. To this end, we introduce angular constraints

$$c_{\text{angular}}(\mathbf{q}) = (\theta_B(\mathbf{q}) - \theta_A(\mathbf{q})) - \hat{\theta}_{\mathbf{i}} \ , \tag{8}$$

where $\hat{\theta}_{\mathbf{i}}$ denotes the target angle specified by the user. We add these angular constraints to the joint constraints $\mathbf{C}$ and compute the target configurations $\mathbf{q}_i$ by minimizing (7) as before. If the resulting minimum is nonzero, we notify the user that the target angles are conflicting with mechanical constraints. Once the target configurations have been computed, they remain fixed throughout the remainder of the design process.

C. Optimization

With the target configurations defined, the goal is now to find spring parameters $\mathbf{p}$ such that the mechanism is statically balanced. To this end, we start by defining the total potential energy for a given pose as

$$U(\mathbf{p}, \mathbf{q}_i) = E_{\text{gravity}}(\mathbf{q}_i) + E_{\text{spring}}(\mathbf{p}, \mathbf{q}_i) \ . \tag{9}$$

The gravitational and elastic components are defined as

$$E_{\text{gravity}}(\mathbf{q}_i) = -\sum_j^{n_c} m_j \mathbf{z}_j \cdot \mathbf{g} \ , \quad \text{and} \tag{10}$$

$$E_{\text{spring}}(\mathbf{p}, \mathbf{q}_i) = \frac{1}{2}\sum_j^{n_s} k_j \left(\|\mathbf{x}_{B,j} - \mathbf{x}_{A,j}\|_2 - l_j\right)^2 \ , \tag{11}$$

where $\mathbf{g}$ is the gravitational acceleration, m_j is the total mass of link $\mathbf{a}_j$, and $\mathbf{z}_j$ its position.

A direct approach to static balance would be to ask that the net generalized forces vanish for every target pose, i.e.,

$$\mathbf{F}(\mathbf{p}, \mathbf{q}_i) = -\frac{\partial U(\mathbf{p}, \mathbf{q}_i)}{\partial \mathbf{q}_i} = \mathbf{0} \quad \forall i \ . \tag{12}$$

Solving these nonlinear equations directly is, however, not an attractive option from an algorithmic point of view: measuring progress is difficult in the absence of a proper objective function, and the over-determined nature of the system leads to additional overhead.

In order to avoid these problems, we leverage the observation that, if the total potential energy is constant throughout target space, then its gradient must vanish at every point; see also Gallego et al. [5]. Instead of requiring force equilibrium, we therefore ask that the total potential energy be the same for all target poses, i.e.,

$$U(\mathbf{p}, \mathbf{q}_i) = \text{const.} \quad \forall i \ . \tag{13}$$

We note that the conditions for force equilibrium and constant energy are only equivalent if they hold point-wise, i.e., for every configuration in target space. While enforcing constant energy for each target pose $\mathbf{q}_i$ does not guarantee force balance in the strict sense, we have never observed a case of constant potential with force imbalance during our experiments.

To implement the constant-energy approach, we must know the value of the energy. However, since this value is not known *a priori*, we construct an objective function that attracts the potential of the individual poses $U(\mathbf{p}, \mathbf{q}_i)$ to their mean $\overline{U}(\mathbf{p})$ as

$$f_{\text{balance}}(\mathbf{p}) = \frac{1}{2}\sum_i^{n_a} \left(U(\mathbf{p}, \mathbf{q}_i) - \overline{U}(\mathbf{p})\right)^2 \ , \tag{14}$$

$$\overline{U}(\mathbf{p}) = \frac{1}{n_a}\sum_i^{n_a} U(\mathbf{p}, \mathbf{q}_i) \ . \tag{15}$$

This leads to an unconstrained minimization problem for the unknown spring parameters $\mathbf{p}$—the configurations $\mathbf{q}_i$ are fixed—that we solve using Newton's method. A zero value for the solution indicates a valid, statically balanced design. However, such a solution does not necessarily exist as explained next.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929984, IEEE Robotics and Automation Letters

D. Spring Reduction

Whether there exists a design with zero balance objective $f_{\text{balance}}(\mathbf{p})$ depends on the selected spring topology, i.e., the number of springs and the links that they connect. Finding a sparse and valid topology that permits zero balance energy with only few springs is not an easy task in general. To support the user in this process, we introduce a sparsity objective that, starting from an exhaustive set of springs that includes all pairwise link connections, gradually eliminates those springs that are least required. To this end, we define an L_1-norm sparsity objective

$$f_{\text{sparsity}}(\mathbf{p}) = \|\mathbf{k}\|_1 = \sum_i^{n_s} |k_i| \ , \tag{16}$$

where $\mathbf{k} = (k_1, k_2, ..., k_{n_s})^T$ gathers per-spring stiffness coefficients k_i. We then combine sparsity and static balance into a single objective function

$$f(\mathbf{p}) = \alpha f_{\text{balance}}(\mathbf{p}) + (1-\alpha) f_{\text{sparsity}}(\mathbf{p}) \ , \tag{17}$$

where α is a user-provided weight (we use $\alpha = 0.5$). The combined objective is augmented with a set of bound constraints that prevent initial lengths and the stiffness values from becoming negative. We solve the corresponding optimization problem

$$\min_{\mathbf{p}} \quad f(\mathbf{p}) \quad \text{s.t.} \quad l_i \geq 0, \ k_i \geq 0 \quad \forall i \ , \tag{18}$$

using sequential quadratic programming (SQP). If we detect a sufficiently small k_i when $f(\mathbf{p}) = 0$, we remove the corresponding spring and solve again. Once no further spring can be removed, we compute the final design parameters by solving (18) again using only the balance objective.

E. Null-Space Exploration

For a given spring topology, there generally exists a space of parameters satisfying the static balance conditions. Some solutions from this space may be more desirable than others in terms of functional or aesthetic goals. For example, the user may want to avoid designs with spring attachment points far away from links, as shown in Fig. 4. Instead of introducing artificial regularizers to favor certain solutions, we allow the user to interactively explore the space of feasible designs and select the one that they prefer. We refer to this process as *null-space exploration*.

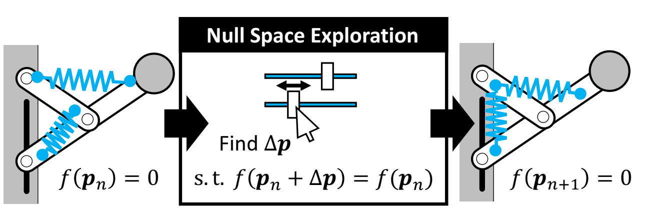


Fig. 4. Null-space exploration. The user manipulates specific spring parameters by dragging sliders. All other parameters are automatically adjusted to ensure that the mechanism remains statically balanced.

To guarantee design feasibility during null-space exploration, we require that any parameter change $\Delta\mathbf{p}$ leave the balance objective and its gradient unchanged, i.e.,

$$f(\mathbf{p}+\Delta\mathbf{p}) = 0 \quad \text{, and} \quad \frac{\partial}{\partial \mathbf{p}} f(\mathbf{p}+\Delta\mathbf{p}) = 0 \ . \tag{19}$$

Expanding the second expression around $\mathbf{p}$, we obtain a first-order condition for maintaining optimality,

$$\frac{\partial^2 f(\mathbf{p})}{\partial \mathbf{p}^2} \Delta\mathbf{p} = \mathbf{H}\Delta\mathbf{p} = \mathbf{0} \ , \tag{20}$$

where $\mathbf{H}$ is the Hessian of the balance objective. The above condition is equivalent to requiring that any admissible parameter change $\Delta\mathbf{p}$ must lie in the null-space of $\mathbf{H}$. To enforce this condition, we construct a null-space basis $\mathbf{Z}$ of $\mathbf{H}$ using eigenvalue decomposition and require that $\Delta\mathbf{p} = \mathbf{Z}\mathbf{w}$ for some $\mathbf{w} \in \mathbb{R}^m$, where m is the number of zero eigenvalues.

During null-space exploration, the user interactively adjusts spring parameters using a set of sliders as illustrated in Fig. 4. These adjustments give rise to a target change $\Delta\bar{p}_i$ for a given parameter p_i. We project the desired change to the null-space of $\mathbf{H}$ by solving the minimization problem

$$\min_{\mathbf{w}} \ \beta \|\mathbf{Z}\mathbf{w}\|_2^2 + (1-\beta)\left[(\mathbf{Z}\mathbf{w})_i - \Delta\bar{p}_i\right]^2 \ , \tag{21}$$

where β balances between the conflicting goals of achieving the desired parameter change and minimizing the change in the remaining parameters.

F. Implementation

We implemented our method in C++ using Eigen[1] for linear algebra and OpenSiv3D[2] for visualization. All of our algorithms are sufficiently fast to run at interactive rates on a standard desktop PC (2.8GHz, single-threaded) for all examples presented in this work.

IV. RESULTS

We evaluated our method on a set of simulation examples and several physical prototypes. The first set of experiments aims at verifying basic properties, while the second set demonstrates the potential of our method to create novel and complex statically balanced mechanisms. The design time for the examples that we show ranged from a few minutes for simple mechanisms (e.g., Fig. 5) to just under half an hour for the most complex ones (Fig. 8, (b)). The largest fraction of this time was spent on exploring design variations for different spring topologies—all computations are interactive.

A. Validation

To validate that our method is able to produce designs known from the literature, we apply it to the gravity equilibrator shown in Fig. 1a. The parameters computed using our method correspond to the analytical solution predicted by Eq. 1. Using null-space exploration, users can quickly create design variations with different spring attachment points, as illustrated in Fig. 5. We furthermore verified numerically through null-space exploration that the only possible value for the initial length of the spring is indeed zero.

[1]https://eigen.tuxfamily.org
[2]https://github.com/Siv3D/OpenSiv3D

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929984, IEEE Robotics and Automation Letters

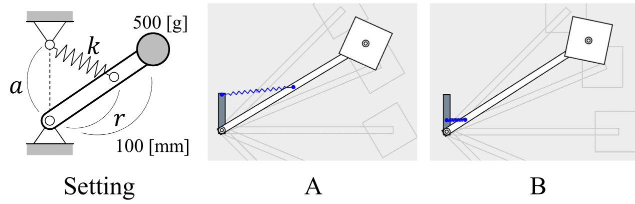


Fig. 5. Two design variations (*middle*, *right*) for the gravity equilibrator (*left*) generated with our method.

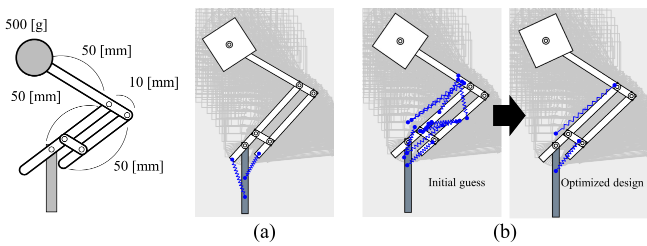


Fig. 6. Automatic spring topology design illustrated on the Anglepoise mechanism (*left*). Starting from an exhaustive set of springs, our method discovers an alternative solution with two springs through automatic sparsification.

To assess the effectiveness of our sparsity objective, we revisit the well-known Anglepoise mechanism shown in Fig. 6. We start with an exhaustive set of springs as initial guess and perform several iterations of the reduction process described in Sec. III-E. The result is a functional, statically balanced mechanism, shown in Fig. 6 (b), which represents an alternative solution to the original design shown in Fig. 6 (a).

Fig. 7 shows a simple example that illustrates our null-space exploration method. Starting from an initial design with spring attachment points far away from their corresponding links, the user interactively adjusts spring parameters to obtain a more compact design.

B. Design Examples

Our method can be used to quickly generate new variations on existing statically balanced mechanisms, but its true strength lies in the application to mechanisms for which no statically balanced solutions are available. To illustrate the potential of our method for this task, we chose three comparatively complex mechanical designs as shown in Fig. 8. To achieve a desired load-bearing capacity, we apply a weight force of 2.5 [N] to the mechanisms; see Fig. 10. In order to facilitate manufacturing, we use off-the-shelf springs and constrain the corresponding design parameters to the measured stiffness value during optimization. We use PLA filament to 3D-print the rigid links as well as industrial ball bearings and steel shafts for the joints in order to reduce friction. We furthermore used pulleys and cables to adjust the effective free length of the springs [8].

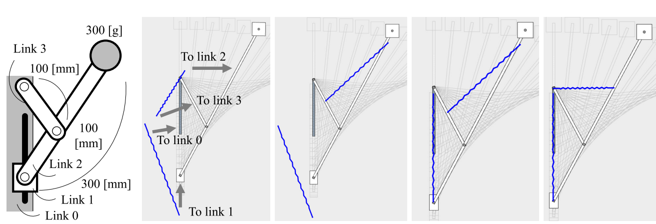


Fig. 7. Using null-space exploration, the user creates a more compact design through a sequence of three parameter adjustment steps.

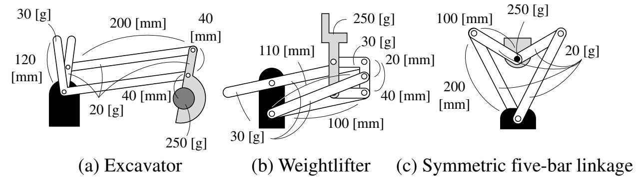


Fig. 8. Three rigidly-articulated mechanisms that were used as input for our method to create statically balanced versions.

The manufactured designs are shown in Fig. 9. As can be seen in the accompanying video, our mechanisms can be moved effortlessly and maintain static equilibrium throughout a wide range of configurations; see also Fig. 10.

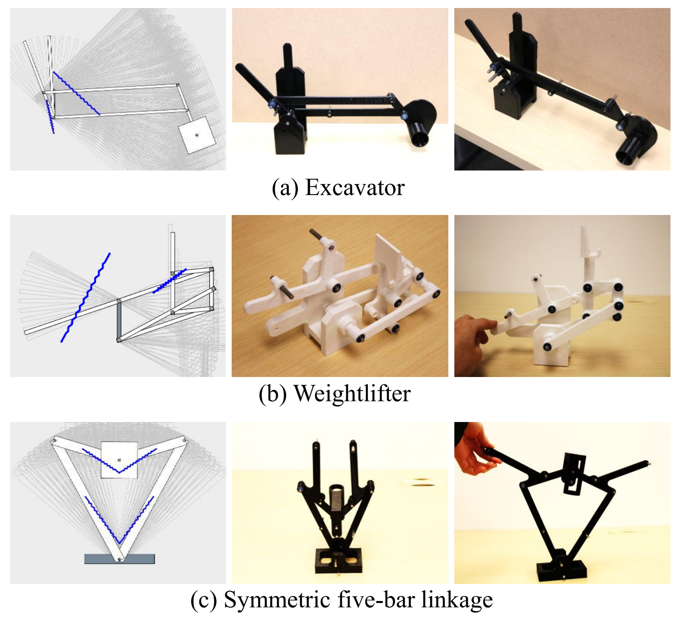


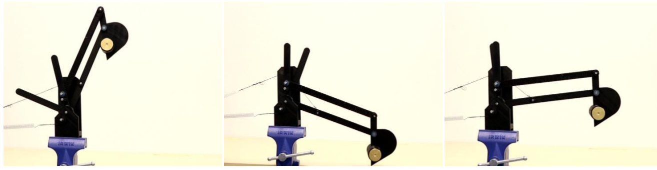
Fig. 9. Optimized designs (*left*) and manufactured mechanisms (*middle*, *right*).

V. DISCUSSION AND LIMITATIONS

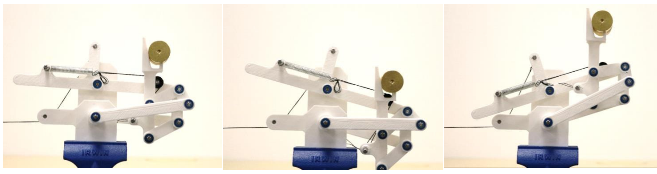
We proposed an interactive, optimization-driven approach for designing statically balanced mechanisms. Our method computes the required number of springs, their attachments points, and initial lengths in a semi-automatic way that integrates user input. Our null-space exploration method is an efficient and effective tool to navigate the local space of design alternatives. While our results suggest that our method is a flexible and powerful approach for creating new statically balanced mechanisms, there are several limitations that we discuss below.

Even when using an exhaustive set of springs, there is no guarantee that a given mechanism will admit a statically balanced design. Our method does also not formally guarantee that the resulting number of springs is the minimum required

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2019.2929984, IEEE Robotics and Automation Letters



(a) Excavator



(b) Weightlifter

(c) Symmetric five-bar linkage

Fig. 10. Snapshots of manufactured mechanisms showing static equilibrium for different poses.

to achieve static balance. Similarly, while our system will converge to a locally-optimal design, there may exist remote solutions that cannot be reached during null-space exploration. While these limitations are to a large extent inherent to nonconvex optimization, our interactive system mitigates these problems by enabling the user to guide the optimization towards new regions of the solution space, if desired.

While we have only considered planar mechanisms in this work, it would be interesting to explore the design of statically balanced spatial mechanisms in the future. This setting gives rise to new challenges, in particular mechanical singularities and collisions between elements [27].

When manufacturing our examples, our goal was to minimize friction in the joints. While minimal friction implies minimal effort when transitioning between configurations, it might be interesting to allow for finite friction in the design to increase the region of static balance.

Lastly, our method does not support changes in the underlying mechanism during optimization. Using such changes as degrees of freedom during optimization might lead to more flexibility and, ultimately, better designs.

References

[1] J. Lasseter, "Principles of traditional animation applied to 3D computer animation," *ACM Siggraph Comp. Graph.*, vol. 21, no. 4, pp. 35–44, 1987.

[2] M. French and M. Widden, "The spring-and-lever balancing mechanism, George Carwardine and the Anglepoise lamp," *Journal of Mechanical Engineering Science*, vol. 214, no. 3, pp. 501–508, 2000.

[3] Y. R. Chheta, R. M. Joshi, K. K. Gotewal, and M. ManoahStephen, "A review on passive gravity compensation," in *Proc. of ICECA 2017*, pp. 184–189, April 2017.

[4] J. L. Herder, *Energy-free Systems. Theory, conception and design of statically*. PhD thesis, Delft University of Technology, 2001.

[5] J. A. Gallego and J. L. Herder, "Criteria for the static balancing of compliant mechanisms," in *ASME 2010 IDETC/CIE*, pp. 465–473, American Society of Mechanical Engineers, 2010.

[6] G. Radaelli, J. A. Gallego, and J. L. Herder, "An energy approach to static balancing of systems with torsion stiffness," *Journal of Mechanical Design*, vol. 133, no. 9, p. 091006, 2011.

[7] G. Radaelli and J. L. Herder, "Isogeometric shape optimization for compliant mechanisms with prescribed load paths," in *ASME 2014 IDETC/CIE*, 2014.

[8] T. Morita, F. Kuribara, Y. Shiozawa, and S. Sugano, "A novel mechanism design for gravity compensation in three dimensional space," in *Proc. of IEEE/ASME AIM*, pp. 163–168, 2003.

[9] J. P. Whitney and J. K. Hodgins, "A passively safe and gravity-counterbalanced anthropomorphic robot arm," in *Proc. of IEEE ICRA*, pp. 6168–6173, 2014.

[10] V. Arakelian, "Gravity compensation in robotics," *Advanced Robotics*, vol. 30, no. 2, pp. 79–96, 2016.

[11] Y. Chen, J. Fan, Y. Zhu, J. Zhao, and H. Cai, "A passively safe cable driven upper limb rehabilitation exoskeleton," *Technology and Health Care*, vol. 23, no. s2, pp. S197–S202, 2015.

[12] H.-C. Hsieh, L. Chien, and C.-C. Lan, "Mechanical design of a gravity-balancing wearable exoskeleton for the motion enhancement of human upper limb," in *Proc. of IEEE ICRA '15*, pp. 4992–4997, IEEE, 2015.

[13] P. Robertson, C.-H. Kuo, and J. Herder, "A compatibility study of static balancing in reconfigurable mechanisms," in *ASME IDETC/CIE '16*, 2016.

[14] A. Garg, A. Jacobson, and E. Grinspun, "Computational Design of Reconfigurables," *ACM Trans. Graph.*, vol. 35, no. 4, 2016.

[15] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, "Computational Design of Mechanical Characters," *ACM Trans. Graph.*, vol. 32, no. 4.

[16] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross, "Computational Design of Linkage-based Characters," *ACM Trans. Graph.*, vol. 33, pp. 64:1–64:9, July 2014.

[17] M. Bächer, S. Coros, and B. Thomaszewski, "LinkEdit: Interactive Linkage Editing Using Symbolic Kinematics," *ACM Trans. Graph.*, vol. 34, pp. 99:1–99:8, July 2015.

[18] V. Megaro, J. Zehnder, M. Bächer, S. Coros, M. Gross, and B. Thomaszewski, "A Computational Design tool for Compliant Mechanisms," *ACM Trans. Graph.*, vol. 36, pp. 82:1–82:12, July 2017.

[19] V. Megaro, E. Knoop, A. Spielberg, D. I. W. Levin, W. Matusik, M. Gross, B. Thomaszewski, and M. Bächer, "Designing Cable-driven Actuation Networks for Kinematic Chains and Trees," in *Proc. of the ACM SIGGRAPH / Eurographics SCA '17*, 2017.

[20] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, no. 0278364918771172, 2018.

[21] M. Lustig, A. Dunning, and J. Herder, "Parameter analysis for the design of statically balanced serial linkages using a stiffness matrix approach with cartesian coordinates," in *14th IFToMM world congress*, pp. 122–129, 2015.

[22] S. R. Deepak and G. Ananthasuresh, "Perfect static balance of linkages by addition of springs but not auxiliary bodies," *Journal of mechanisms and robotics*, vol. 4, no. 2, p. 021014, 2012.

[23] P. Lin, W. Shieh, and D. Chen, "Design of statically balanced planar articulated manipulators with spring suspension," *IEEE Transactions on Robotics*, vol. 28, pp. 12–21, Feb 2012.

[24] Y.-Y. Lee and D.-Z. Chen, "Determination of spring installation configuration on statically balanced planar articulated manipulators," *Mechanism and Machine Theory*, vol. 74, pp. 319 – 336, 2014.

[25] E. G. Merriam, M. Colton, S. Magleby, and L. L. Howell, "The design of a fully compliant statically balanced mechanism," in *ASME IDETC/CIE '13*, 2013.

[26] G. Radaelli and J. L. Herder, "Shape optimization and sensitivity of compliant beams for prescribed load-displacement response," *Mechanical Sciences*, vol. 7, no. 2, pp. 219–232, 2016.

[27] C. Gosselin, "Gravity compensation, static balancing and dynamic balancing of parallel mechanisms," in *Smart Devices and Machines for Advanced Manufacturing*, pp. 27–48, Springer, 2008.