2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Computational Design of Balanced Open Link Planar Mechanisms with Counterweights from User Sketches

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

Abstract— **We consider the design of under-actuated articulated mechanism that are able to maintain stable static balance. Our method augments an user-provided design with counterweights whose mass and attachment locations are automatically computed. The optimized counterweights adjust the center of gravity such that, for bounded external perturbations, the mechanism returns to its original configuration.**

Using our sketch-based system, we present several examples illustrating a wide range of user-provided designs can be successfully converted into statically-balanced mechanisms. We further validate our results with a set of physical prototypes.

I. INTRODUCTION

Designing articulated mechanisms that fulfill both functional and aesthetic goals is challenging problem at the intersection of art and engineering. A prominent set of examples is “Karakuri Ningyo[1],” which comprises Japanese automata from the Edo period in the 18^{th} century; see [1]. A particularly noteworthy design in this set is “Chahakobi Ningyo,” a mechanical character that serves tea to the audience using the weight of the tea itself for balancing. Another example is “Yumihiki Doji,” by Hisashige Tanaka, which can shoot a bow by moving each articulated joint.

When such an articulated mechanism is driven by an actuator, it is important to consider the weight of the various mechanical components. If the design is inadequate, the torque required to support the structure may cause the mechanism to vibrate or to move incorrectly. For this reason, self-weight compensation strategies that reduce the static loads that must be supported by actuators have been studied extensively [2], [3].

On the other hand, underactuated mechanisms such as “roly-poly toys” can effortlessly balance even when the posture changes due to external force [4]. This stable balancing can be achieved by adjusting the center of gravity as a function of posture. One commonly used means to this end are counterweights. While solutions exist for the single-component case [4], designing counterweights that achieve static balance for underactuated multi-component mechanisms is a problem for which no effective solution has yet been described.

[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. bernhard@iro.umontreal.ca

[1]“Karakuri Puppet” on Wikipedia

In this work, we propose a method that automatically determines the parameters of counterweights such that, when added to a user-provided input design, allows the corresponding under-actuated mechanism to maintain its balance and to return to this configuration upon bounded perturbations. In addressing this problem, we are guided by three main requirements. should meet the following three requirements simultaneously:

- *Multi-component mechanism:* the mechanism is composed of multiple rigid bodies and joints.
- *Sketch-based input:* the initial design is provided by the user through a sketch-based interface.
- *Stable balancing:* the final mechanism maintains its balance in the target pose, and automatically returns to this pose after bounded perturbation.

As a first step towards this goal, we develop a system that generates an open-link planar balanced mechanism from a user-specified sketch and semi-automatically discovers the counterweight shapes using an evolutionary optimization strategy. We demonstrate our method on a set of virtual designs that are created interactively, entirely within our system. To validate the feasibility of our designs, we furthermore built several physical mechanisms.

II. RELATED WORK

Our method builds on previous work from three main research domains: mechanical character design, statically balanced mechanisms, and optimization-based design of statically-stable objects. The three domains are discussed in the following subsections.

A. Mechanical Character Design

Fueled by the increasing availability of additive manufacturing technologies, the computer graphics community has started to embrace the design of 3D-printable objects, including functional mechanisms with desired aesthetics computer graphics community [5], [6]. Our work draws particular inspiration from the work by Baecher et al. [7] that automatically converts a digital character into components such that, once printed, the physical character can be posed within a desired range of motion. Our work is also in line with recent methods for creating animated mechanical characters [8], [9], which we aim to extend to the problem of static balance.

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

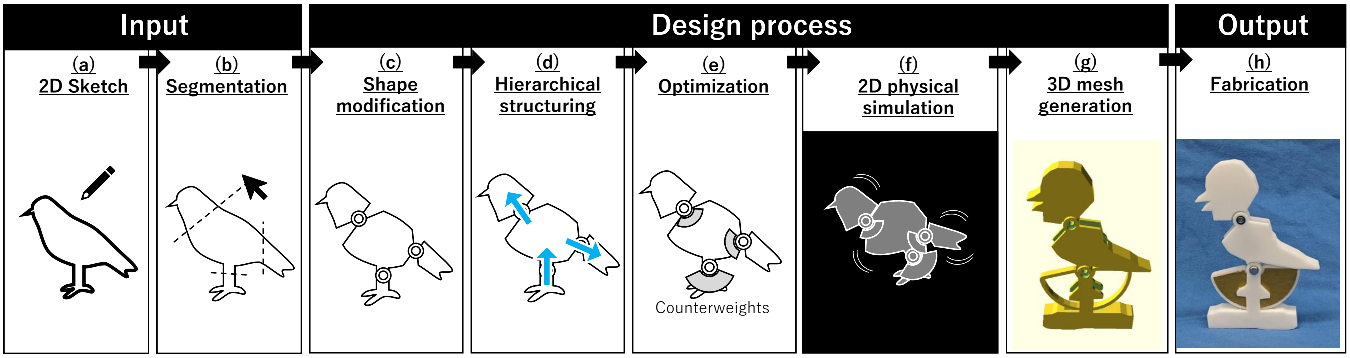

Fig. 1. Overview of our design system. A user draws a sketch and specifies the location for division. The system then optimizes the parameters of counterweights based on physical simulation. The final design is converted into a set of 3D printable meshes ready for fabrication.

B. Statically Balanced Mechanism

Many works in robotics and mechanics have investigated gravity compensation using counter-weights and compliant elements such as springs [2], [3]. For example, Takahashi et al. [10] proposed an interactive design system that automatically suggests spring parameters to achieve static balance throughout user-specified space of configurations. However, they did not consider the design of counterweights. Several works proposed methods to design counterweights for specific mechanisms [11], [12], [13]. In this work, we aim to develop a system that automatically adjusts counterweight parameters to achieve static balance, even for a new mechanisms specified by the user.

C. Optimization Based Stable Object Design

In the context of design for additive manufacturing, a recent stream of research has addressed the problem of optimizing the center of gravity of a given input shape [14], [15], [16]. In a similar vein, Zhao et al. [4] proposed a design tool for so called *roly-poly* toys that can return to the correct posture by themselves when they are pushed over. To our knowledge, adjusting the center of gravity to achieve static balance for articulated mechanisms is a problem that has not yet been addressed.

III. METHOD

As illustrated in Fig. 1, the workflow of our system consists of eight steps. First, the user-provided sketch is converted into a 2D polygon and joint positions are automatically determined from the cutting lines specified by the user. Subsequently, link shapes are modified and a hierarchical structure is created as explained in Section III-D. The system subsequently performs optimization to calculate the counterweight shapes. Physical simulation is performed using the optimized shapes to verify that the result is balanced. Once static balance is confirmed, the system generates 3D shapes by Boolean operations, as explained in Section III-E. Finally, the output shapes are 3D printed and connected through off-the-shelf bearings and shafts.

A. Mechanism

We represent articulated mechanisms using a hierarchical structure composed of n_l links and n_j joints (see Fig. 2). The links represent rigid bodies and the joints represent the constraints on the relative position of two links. One of the links in the mechanism must connect to the fixed root link, which represents the top of the hierarchical structure. We exclusively consider designs that can be described in this way, which precludes mechanisms with closed loops.

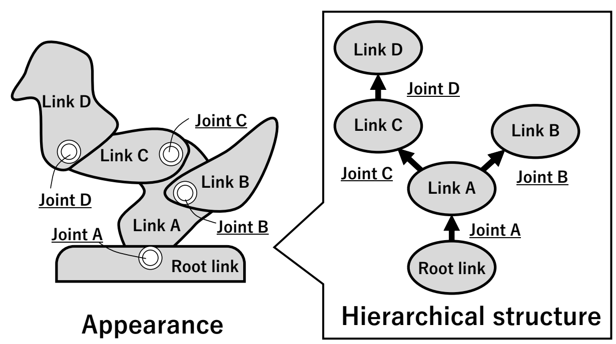

Fig. 2. Diagram of the hierarchical structure used to represent mechanisms. A mechanism is composed of links and joints. The joints define the positional relationships between a given child link and its parent link.

A link constitutes an articulated mechanism, and it can change its position and angle in the planar condition. A link also contains 2D shape information itself as a rigid body. The 2D shape information is a vector of vertices in local coordinates, which we refer to as a 2D polygon. Therefore, the links can be described as

$$\mathbf{l}_i = \left(\mathbf{p}^T, \theta, \hat{\mathbf{s}}^T\right)^T \quad , \tag{1}$$

where $\mathbf{p}$ is the 2D position in global coordinates, θ is an angle, and $\hat{\mathbf{s}}$ is a vector representing a 2D polygon.

A joint restricts the positional relationship between two links. We focus on pin joints and formulate corresponding position constraints as

$$\mathbf{j}_i = \left(\hat{\mathbf{p}}^T_{\text{parent}}, \hat{\mathbf{p}}^T_{\text{child}}, \hat{\theta}_{\text{range}}\right)^T \quad , \tag{2}$$

where $\hat{\theta}_{\text{range}}$ represents the movable range of the relative angle between a given child link and its parent link, and

$\hat{\mathbf{p}}^T_{\text{parent}}$ and $\hat{\mathbf{p}}^T_{\text{child}}$ are joint positions in the corresponding local coordinate frames.

B. Counterweights

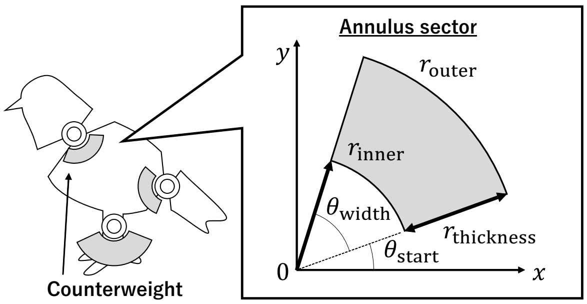

Fig. 3. We use counterweights in the form of annulus sectors. These shapes are defined by four parameters as indicated.

A counterweight structure consists of n_w weights, each of which is attached to a movable link. To obtain a low-dimensional parameterization, we restrict the space of counterweight shapes to annulus sectors, which are defined by two radii (r_{inner}, r_{outer}) and angles (θ_{start}, θ_{width}); see Fig. 3. A counterweight connected to the i-th link can be defined as

$$\mathbf{w}_i = (r_{\text{inner}}, r_{\text{thickness}}, \theta_{\text{start}}, \theta_{\text{width}})^T \quad , \tag{3}$$

where $r_{\text{thickness}}$ is the difference between the inner radius r_{inner} and the outer radius r_{outer} of an annulus sector.

C. Optimization

The goal for the optimization stage is to find parameters for all counterweights such that the articulated mechanism is in static equilibrium in the target configuration. We summarize the parameters $\mathbf{x}$ to be optimized as

$$\mathbf{x} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{n_w})^T \, . \tag{4}$$

When computing the center of gravity for a given link in the mechanism, the weight of all children must be considered when. To this end, we recursively calculate the center of gravity of all links in the hierarchical structure as follows:

$$\mathbf{p}_i^{\text{CoG}} = \frac{\sum_{j \in \mathscr{I}_i} (\rho_{\text{l}} A(\mathbf{l}_j)\, \mathbf{c}(\mathbf{l}_j) + \rho_{\text{w}} A(\mathbf{w}_j)\, \mathbf{c}(\mathbf{w}_j))}{\sum_{j \in \mathscr{I}_i} (\rho_{\text{l}} A(\mathbf{l}_j) + \rho_{\text{w}} A(\mathbf{w}_j))}, \tag{5}$$

where $\mathscr{I}_i$ represents the set of indices for the i-th link and its children, the function A calculates the area of a 2D polygon, the function $\mathbf{c}$ calculates the centroid of a 2D polygon, and ρ_{l} and ρ_{w} are mass densities for the link-material and the counterweights, respectively.

If the line connecting the base joint of a given link with its center of gravity (including contributions from all children) is not aligned with the direction of gravity, a nonzero moment occurs and the structure is not in static balance. To achieve balance, we therefore define the objective function for computing counterweight parameters as follows:

$$f(\mathbf{q}) = \sum_i \|\mathbf{p}_i^{\text{CoG}} - \left(\mathbf{p}_{\text{i,parent}} + \alpha \frac{\mathbf{g}}{\|\mathbf{g}\|}\right)\|^2 \ , \tag{6}$$

$$\min_{\mathbf{q}} \quad f(\mathbf{q}) \quad \text{s.t.} \quad r_{\text{thickness},i} \geq 0, \ \theta_{\text{width},i} \geq 0 \quad \forall i \ , \tag{7}$$

where $\mathbf{q}$ is a vector that accumulates all of the mechanical information as $\mathbf{q} = \left(\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_{n_l}, \mathbf{j}_1, \mathbf{j}_2, ..., \mathbf{j}_{n_j}, \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{n_w}\right)^T$, $\mathbf{p}_{\text{i,parent}}$ represents the (bottom) joint position of the parent, $\mathbf{g}$ is gravitational acceleration in global coordinates, and α is the target distance from the parent joint (we use $\alpha = 5$ [mm]). We solve this nonlinear minimization problem using CMA-ES [17], [18], a derivative-free method for numerical optimization.

Following the optimization, the system performs 2D physics simulation on the optimized results to rule out designs that are not stable. We use the impulse-based solver [19], which achieves real-time rates for all examples shown in this work.

D. 2D Sketch Interface

Our system allows the user to create open-link mechanisms simply from a sketch, as illustrated in Fig. 4.

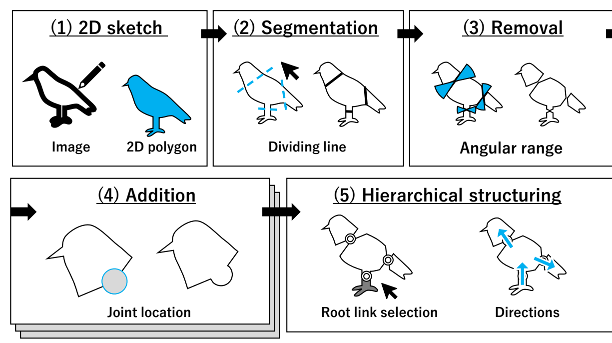

Fig. 4. Diagram of process from 2D sketch to mechanism definition. The system polygonizes the sketch drawn by the user (1) and splits it according to the dividing lines (2). Thereafter, the system cuts the links so that the joint admits angle changes in the desired range (3).Subsequently, geometry is added for attaching joints (4). Then, the system sets up the link hierarchy using the user-selected root link (5).

First, the system extracts contours from a user-drawn rasterized sketch and converts these into 2D polygons. A simplification method known as the Ramer–Douglas–Peucker algorithm [20], [21] is used for converting the rasterized image into 2D polygons.

To segment a 2D polygon into an articulated mechanism with links and joints, the user draws a line on the shape to decide where to divide it. The position of the joint is automatically set to the center of the segmented section. The 2D polygon is then cut based on the desired angular range of motion $\hat{\theta}_{\text{range}}$.

Moreover, the system adds a circular shape to every link in the position where joints will be placed. This added geometry is used for placing mechanical joint parts such as shafts and bearings. We set its radius to $r_{\text{joint}} = 5mm$.

Once the system has modified the shape of the links, it defines the hierarchical relationships by determining the joints and their corresponding link IDs that correspond to the parent and child links. The user specifies the root link, upon which these link IDs of the joints are determined automatically according to the distance from the root link.

E. 3D Mesh Generation

Our system translates optimized 2D shapes into 3D mesh data, as illustrated in Fig. 5, using 3D Boolean operations. To avoid collisions between links and counterweights, we design 3D meshes with two layers. The thickness for the inner layer is a_{inner} and the outer layer thickness on each side is $a_{\text{outer}} = a_{\text{inner}}/2$. Furthermore, the clearance between the layers is $a_{\text{clearance}}$ (we use 1 [mm]).

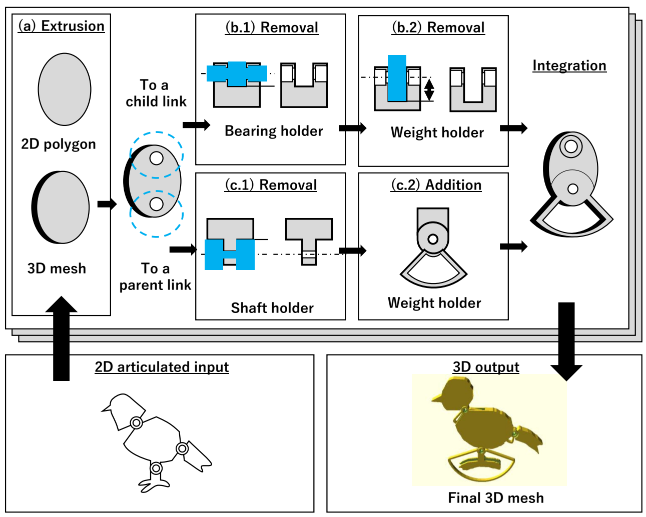

Fig. 5. Process from mechanism definition to 3D mesh output. The system performs 3D Boolean operations.

First, we extrude the 2D polygons of the link with the thickness $a_{\text{extrude}} = a_{\text{inner}} + 2a_{\text{outer}} + 2a_{\text{clearance}}$, as indicated in Fig. 5(a). Around the joint connected to the child side (Fig. 5(b)), we cut the inner layer with a radius of $r_{\text{cut}} = r_{\text{joint}} + a_{\text{clearance}}$ or the corresponding counterweight's outer radius $r_{\text{outer}} + a_{\text{clearance}}$. To fit the bearings, we also create a hole with diameter d_{bearing} in the outer layer. However, around the joint connected to the parent side (Fig. 5(c)), we cut the outer layer with a radius of r_{cut} and the inner layer with a diameter of d_{shaft}. Thereafter, we add the counterweight fixtures by adding an extruded polygon in the shape of the counterweight. By performing these operations on every link, we obtain a 3D-printable mesh.

F. Implementation

For the optimization, we use the linear algebra libraries Eigen[2] and libcmaes[3] with C++. To visualize the mechanism, we use OpenSiv3D[4], where the contour extraction function of OpenCV[5] is used for extracting the image outline. Boost geometry[6] is used for polygon simplification, Boolean operations, and calculation of the centroid of the polygons in 2D. Furthermore, Box2D[7] is used for 2D physics simulation.

[2]http://eigen.tuxfamily.org
[3]https://github.com/beniz/libcmaes
[4]https://github.com/Siv3D/OpenSiv3D
[5]https://opencv.org/
[6]https://www.boost.org
[7]https://box2d.org

To generate 3D meshes using the Boolean operations on 3D geometric shapes, we use OpenSCAD[8].

IV. FABRICATION AND EVALUATION

We conducted several tests to ensure that the system operates correctly. These tests include both simple examples as well as relatively complex multi-link mechanisms. For further validation, we also tested several physical prototypes. During this evaluation, we used a 3D printer (Prusa MK3S) and PLA filament for the links, and stainless steel shafts (diameter $d_{\text{shaft}} = 3$ [mm]) and bearings ($d_{\text{bearing}} = 6$ [mm]) for the joints. Moreover, we used brass (thickness $a_{\text{extrude}} = 6$ [mm]) for the counterweight material, which was cut using a CNC mill (KitMill RZ420). We used $\rho_l = 0.6$ [g/cm^2] and $\rho_c = 5.1$ [g/cm^2] as the density. During optimization, the inner radius r_{inner} of the counterweights was fixed to 7 [mm] to allow for convenient assembly.

A. Simple mechanisms

Three types of shapes were used for the simpler test cases: a single joint, two serial links, and three branching links. As indicated in Fig. 6, the segmentation was successful for the shapes drawn by the user. Moreover, each mechanism was able to maintain balance in the designated configuration after optimizing counterweights.

B. Complex mechanisms

Subsequently, three relatively complex shapes were adopted: an arm, a cat, and a monster. As for the simple shapes, we succeeded in stabilizing the target configuration of the mechanism as shown in Fig. 7. However, the system also generated several solutions with collisions between counterweights, which were not suitable for manufacturing. If such a collision problem occurs, adding the sum of the areas of overlap between counterweight shapes to the objective function during optimization may improve the final result.

C. Physical mechanisms

For experimental validation, we used the generated 3D meshes to fabricate physical prototypes: a bird, a human, and an arm. As can be observed in the accompanying video, the physical prototypes successfully balance in their target configuration and return to their original posture after moderate perturbations (see Fig. 8).

V. DISCUSSION AND FUTURE WORK

We have presented an approach for creating statically-balanced mechanisms by optimizing counterweight parameters. Our sketch-based interface allows users to express their design intents in a simple and efficient way. We demonstrated our method on several virtual designs and physical prototypes. As initially intended, we found that optimizing the counterweights by focusing on the center of gravity was useful for creating a statically-stable mechanism. Nevertheless,

[8]https://www.openscad.org

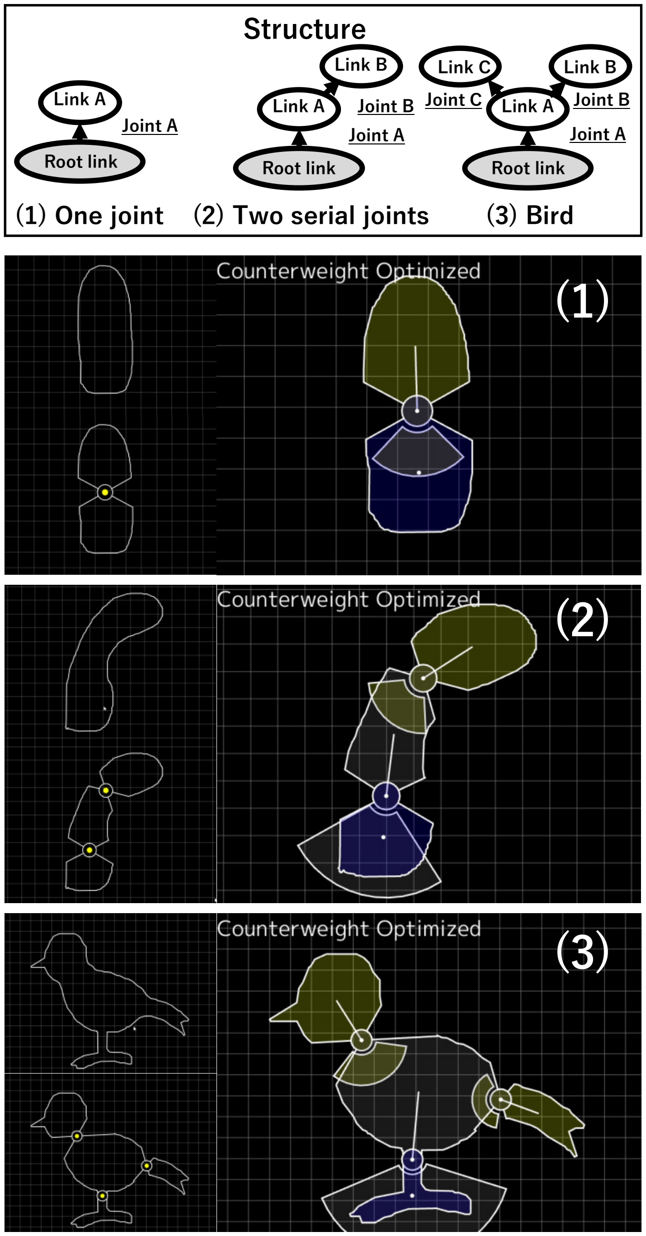

Fig. 6. Results for simple examples. The counterweights attached the joints allowed for stable balance in all cases.

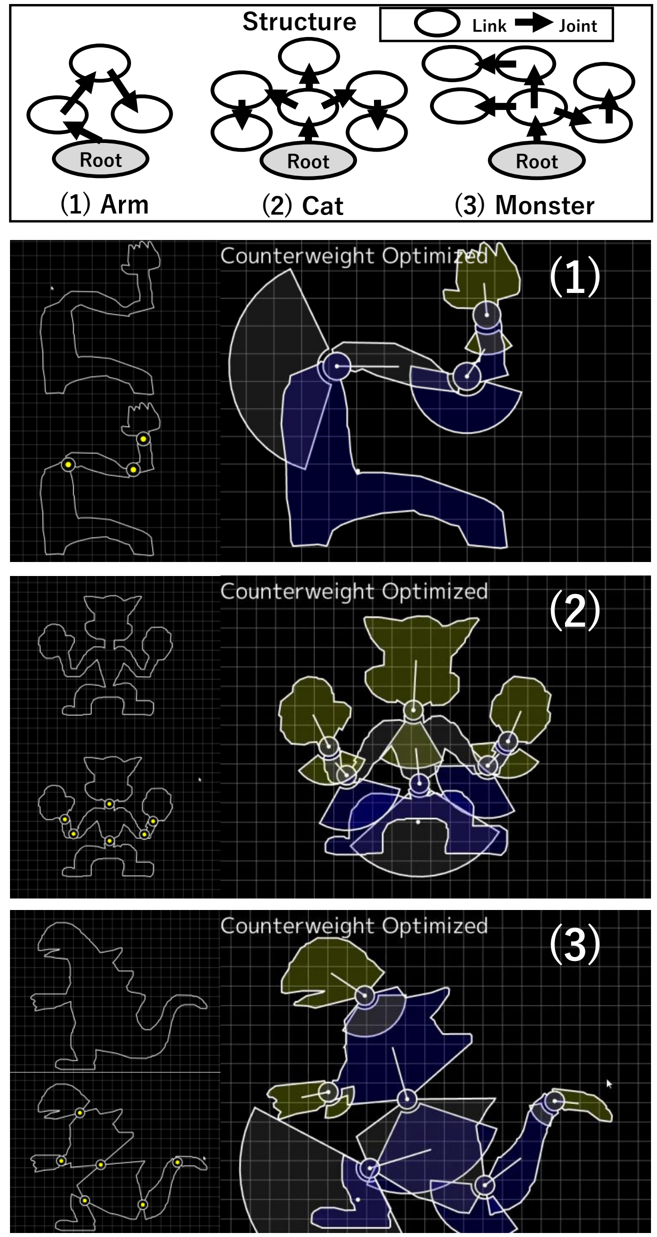

Fig. 7. Result for complex shapes. As for the simple shapes, the mechanisms exhibit stable balance after counterweight optimization. It can be seen that some of the weights are relatively large. Depending on the application, using thicker or higher-density materials for the counterweights might be preferable.

our method currently exhibits various limitations, which we discuss below.

First, the system cannot guarantee that our optimization is able to find a solution in space defined through the user inputs. One solution could be to let the optimization process manage the segmentation of the shape, similar to [7], thus broadening the search space.

Second, the optimization problem might exhibit several optima and a given local solution is in general not globally optimal. In our system, the user cannot modify the parameters following the optimization. However, an iterative process that allows users to tweak parameters could help guiding the system towards a specific solution. An interactive exploration of the solution space using, e.g., the null-space exploration scheme described in [10] might also be helpful.

Third, even if the mechanism is statically balanced, cases exist in which the links and joints collide with each other such that the articulated mechanism cannot even be assembled. For example, the mechanisms in Figs. 7(2) and 7(3) are balanced but collisions occur among the links, joints, and counterweights. When designing a printable layout of the articulated mechanism, a constraint-aware modeling method could aid in solving this problem. See also [8], [22].

In this study, we focused exclusively on open link mechanisms; that is, mechanisms that can be represented using a hierarchical structure. Closed link mechanisms are currently not supported. Finally, our system does not take into account the weight of the joint parts. The 2D center of gravity does

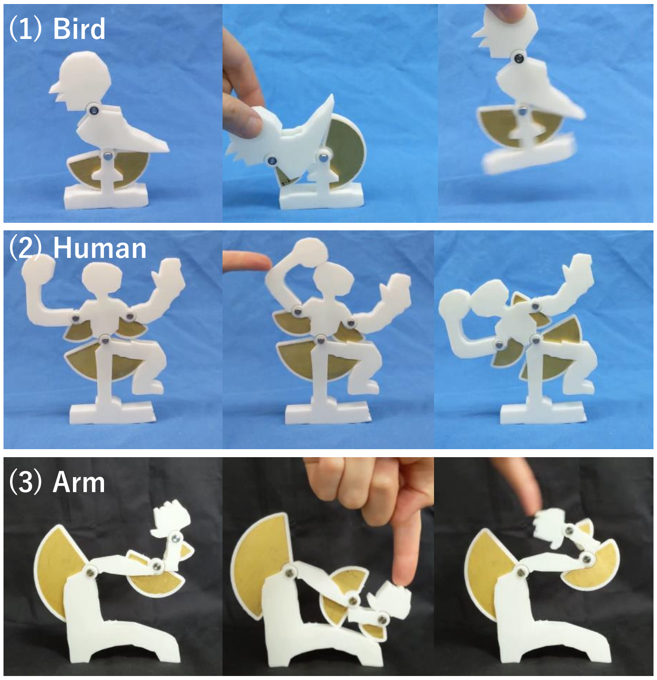

Fig. 8. Physical prototypes for articulated mechanisms designed using the proposed system. After moderate perturbations, each of these mechanisms returns to its stable target configuration.

not always match the 3D printed one. Consequently, several manual modifications are required before physical prototypes can be manufactured. In the future, these aspects should be accounted for during the design process such that the system can adjust the counterweight parameters accordingly.

ACKNOWLEDGMENT

Takuto Takahashi was supported by the Graduate Program for Embodiment Informatics, funded by the JSPS at Waseda University. Hiroshi G. Okuno was supported by Kakenhi 19H00750. Bernhard Thomaszewski was supported by the NSERC Discovery Grants and Discovery Accelerator Awards program.

REFERENCES

[1] Y. Yokota, "A Historical Overview of Japanese Clocks and Karakuri," in *International Symposium on History of Machines and Mechanisms*, pp. 175–188, Springer, 2009.

[2] 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.

[3] C. Yu, Z. Li, and H. Liu, "Research on Gravity Compensation of Robot Arm Based on Model Learning*," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 635–641, July 2019.

[4] H. Zhao, C. Hong, J. Lin, X. Jin, and W. Xu, "Make it swing: Fabricating personalized roly-poly toys," *Computer Aided Geometric Design*, vol. 43, pp. 226 – 236, 2016. Geometric Modeling and Processing 2016.

[5] V. Megaro, B. Thomaszewski, D. Gauge, E. Grinspun, S. Coros, and M. Gross, "ChaCra: An Interactive Design System for Rapid Character Crafting," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '14, (Goslar, DEU), p. 123–130, Eurographics Association, 2015.

[6] Y. Mori and T. Igarashi, "Plushie: An Interactive Design System for Plush Toys," *ACM Trans. Graph.*, vol. 26, p. 45–es, July 2007.

[7] M. Bächer, B. Bickel, D. L. James, and H. Pfister, "Fabricating Articulated Characters from Skinned Meshes," *ACM Trans. Graph.*, vol. 31, pp. 47:1–47:9, July 2012.

[8] 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, July 2013.

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

[10] T. Takahashi, J. Zehnder, H. G. Okuno, S. Sugano, S. Coros, and B. Thomaszewski, "Computational Design of Statically Balanced Planar Spring Mechanisms," *IEEE Robotics and Automation Letters*, vol. 4, pp. 4438–4444, Oct 2019.

[11] C. A. Coello Coello, A. D. Christiansen, and A. H. Aguirre, "Multiobjective design optimization of counterweight balancing of a robot arm using genetic algorithms," in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pp. 20–23, Nov 1995.

[12] J. Wu, T. Li, and L. Wang, "Counterweight optimization of an asymmetrical hybrid machine tool based on dynamic isotropy," *Journal of Mechanical Science and Technology*, vol. 27, 07 2013.

[13] Y. Li, J. Wang, X.-J. Liu, and L.-P. Wang, "Dynamic performance comparison and counterweight optimization of two 3-DOF parallel manipulators for a new hybrid machine tool," *Mechanism and Machine Theory*, vol. 45, pp. 1668–1680, 11 2010.

[14] R. Prévost, E. Whiting, S. Lefebvre, and O. Sorkine-Hornung, "Make It Stand: Balancing Shapes for 3D Fabrication," *ACM Trans. Graph.*, vol. 32, July 2013.

[15] M. Bächer, B. Bickel, E. Whiting, and O. Sorkine-Hornung, "Spin-It: Optimizing Moment of Inertia for Spinnable Objects," *Commun. ACM*, vol. 60, p. 92–99, July 2017.

[16] R. Prévost, M. Bächer, W. Jarosz, and O. Sorkine-Hornung, "Balancing 3D Models with Movable Masses," in *Proceedings of the Conference on Vision, Modeling and Visualization*, VMV '16, (Goslar, DEU), p. 9–16, Eurographics Association, 2016.

[17] N. Hansen, "The CMA evolution strategy: a comparing review," *Towards a new evolutionary computation*, pp. 75–102, 2006.

[18] N. Hansen, "The CMA evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[19] M. Tamis, "Comparison between Projected Gauss Seidel and Sequential Impulse Solvers for Real-Time Physics Simulations," 2015.

[20] D. H. Douglas and T. K. Peucker, "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature," *The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[21] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer graphics and image processing*, vol. 1, no. 3, pp. 244–256, 1972.

[22] J. Hergel and S. Lefebvre, "3D fabrication of 2D mechanisms," *Computer Graphics Forum*, vol. 34, no. 2, pp. 229–238, 2015.