

©Copyright 2022

Liang He

Fabricating Kinetic Objects with
3D Printable Spring-Based Mechanisms for Interactivity

Liang He

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Jon E. Froehlich, Chair

Jennifer Mankoff

Adriana Schulz

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science & Engineering

University of Washington

Abstract

Fabricating Kinetic Objects with
3D Printable Spring-Based Mechanisms for Interactivity

Liang He

Chair of the Supervisory Committee:

Jon E. Froehlich

Paul G. Allen School of Computer Science & Engineering

Emerging 3D printing technology promises the rapid creation of physical shapes. With increasingly accessible and low-cost consumer-grade 3D printers, end-users can create custom objects with diverse materials (*e.g.*, rigid plastics and elastic resins) and 3D printing processes such as Fused Deposition Modeling (FDM) and Stereolithography (SLA). However, 3D-printed objects are typically static with limited or no moving parts. Creating 3D printable objects with kinetic behaviors such as deformation and motion is inherently challenging. For example, designing 3D kinematic models requires expert knowledge of mechanical mechanisms, and assembling movable 3D-printed parts is error-prone. To address these problems and enrich the literature for making movable 3D-printed parts, I introduce a novel design and fabrication approach that uses parametric spring-based mechanisms to augment 3D-printed objects with non-static capabilities, such as deformation and actuation for interactions.

In this dissertation, I first investigate how movable 3D-printed objects are made and how prototyping kinetic objects can benefit from 3D printable springs through a large-scale analysis of kinetic creations on *Thingiverse*. Then, I develop a series of novel design techniques and tools for end-users to design and control 3D printable kinetic objects for interactivity, including physical deformation and haptic feedback, self-propelled motion, and sensing through structural deformation. Finally, I evaluate these techniques and tools

using a wide variety of example applications emphasizing different application domains.

TABLE OF CONTENTS

	Page
List of Figures	iv
List of Tables	x
List of Symbols	xi
Chapter 1: Introduction	1
1.1 Mechanical Springs for 3D Printing	4
1.2 Research Approach	5
1.3 Dissertation Goals and Contributions	6
1.4 Summary of Contributions	9
1.5 Dissertation Organization	10
Chapter 2: Background and Related Work	12
2.1 Spring Physics and Theory	12
2.2 Making 3D Printable Objects with Deformation Behaviors	13
2.3 Actuating 3D-Printed Objects for Interaction	18
2.4 Adding Sensing Capabilities to 3D Printable Objects	21
2.5 Parametric Tools for Designing and Controlling 3D Printable Behaviors	24
2.6 Chapter Summary	26
Chapter 3: Making Things Move: Understanding How 3D Printable Kinetic Ob- jects are Designed and Created	28
3.1 Method	30
3.2 Findings	33
3.3 Discussion: Maker Community <i>V.S.</i> Fabrication Research	43
3.4 Chapter Summary	46
Chapter 4: Ondulé: Creating Deformation Behaviors with 3D Printable Springs	47
4.1 Helical Spring Theory	48

4.2	Mechanical Experiments	50
4.3	Spring Deformation Techniques	54
4.4	Interactive Spring Design Tool	57
4.5	Implementation	59
4.6	Validation through Applications	63
4.7	Limitations	67
4.8	Chapter Summary	69
Chapter 5: Kinergy: Enabling 3D-Printed Objects with Self-Propelled Motion using Springs		
		70
5.1	Motion Types for 3D Printing	72
5.2	Kinetic Units	72
5.3	Kinergy Design Tool	78
5.4	Fabrication	88
5.5	Applications	90
5.6	Limitations	96
5.7	Chapter Summary	98
Chapter 6: FlexHaptics: Creating Custom Haptic Interfaces Using Various Planar Compliant Structures		
		100
6.1	<i>FlexHaptics</i> Modules and Mixing Operators	102
6.2	Technical Evaluation	110
6.3	<i>FlexHaptics</i> Design Editor	111
6.4	Haptic Interface Examples Created with <i>FlexHaptics</i>	113
6.5	Limitations	116
6.6	Chapter Summary	118
Chapter 7: Conclusions and Future Work		
		119
7.1	Contributions	119
7.2	Directions for Future Research	123
7.3	Final Remarks	130
Bibliography		131
Appendix A: The Qualitative Codebook and Analysis Source		
		146
A.1	Category 1: Model Category/Purpose	146
A.2	Category 2: Kinetic Component Design	146

A.3	Category 3: Kinetic Behavior	147
A.4	Category 4: Design Fabrication	147
A.5	Category 5: Social Interaction	148
Appendix B: Helical Spring Theory and Mechanical Experiments for Ondulé		149
B.1	Terminology and Concepts in Helical Spring Theory	149
B.2	Mechanical Experiment 1 Setup and Results	150
B.3	Mechanical Experiment 2 Setup	151
B.4	Mechanical Experiment 3 Setup	151

LIST OF FIGURES

Figure Number	Page	
1.1	There is an increase on the sale of 3D printers from 2007 to 2019 before the pandemic (left). The common 3D printing methods include FDM, SLA, SLS, <i>etc.</i> and 3D printing applications benefit from the advances of 3D printing technology and 3D printable materials (right).	2
1.2	This dissertation focuses on how to use 3D printable spring-based mechanisms to enable deformation, actuation, and sensing behaviors.	10
2.1	Three common spring types and parameters: (a) helical spring, (b) spiral spring, and (c) beam spring.	14
2.2	Articulated and transformable objects made with embedded 3D printable joints and hinges.	15
2.3	Deformable 3D-printed objects created with hinges, metamaterials, and other mechanical elements.	16
2.4	Deformable 3D printable objects made with elastic material and environmentally responsive materials.	18
2.5	Making 3D printable kinetic objects for actuation using external motors and movable 3D-printed structures.	18
2.6	Making 3D printable kinetic objects for actuation using kinematic pairs and pressure-controlled actuator.	20
2.7	3D printable actuators are created for making 3D-printed objects move.	21
2.8	3D printable objects use external sensors or integrate sensing techniques for interaction.	22
2.9	3D printable objects use embedded kinetic structures for sensing.	23
2.10	Applying predefined units from a template library for augmented behavior design in CAD tools.	24
2.11	Direct editing 3D models for desired behaviors in CAD tools.	26
3.1	A set of example 3D-printed kinetic designs found on Thingiverse showcase the usage of integrated 3D-printable kinematic mechanisms such as (a) hinges, (b) gears, and (c) joints, or (d) elastic material, or external hardware such as (e) strings, (f) servos, and (g) electronics for kinetic behaviors.	28
3.2	The iterative process for refining the search term list.	31

3.3	Examples of identical 3D designs in our dataset: (a) a set of remixed bearing designs without additional new designs and (b) different flexible animal figurines with the hinge-based design. The check marks mean the creations are selected to store in our dataset, while the X marks indicate those are not included.	32
3.4	The final dataset consists of designs that use 3D-printable mechanisms, external hardware, and purely elastic materials.	34
3.5	Examples of 3D kinetic designs across 11 application categories defined by Thingiverse.	34
3.6	Examples of 3D printable kinetic mechanisms.	35
3.7	Examples of 3D printable kinetic behaviors.	37
3.8	Examples of actuation methods.	40
4.1	We introduce <i>Ondulé</i> , an interactive tool that allows designers to create and control deformable objects with embedded springs and joints. Above, a workflow shows how to make a solid seahorse body bendable and twistable: (a) select a seahorse body; (b) change the spring length and regenerate the spring directly on the model; (c) control spring stiffness; (d) parameterize spring deformation behaviors by adding additional joints; and (e) print the deformable seahorse with a consumer-grade FDM 3D printer.	47
4.2	Basic helical spring deformation behaviors: (a) compress, (b) extend, (c) twist, and (d) laterally bend.	49
4.3	Mechanical experiment setups: (a) the load frame stretches a 3D-printed rod; (b) the load frame stretches a 3D-printed helical; and (c) the motor rotates a helical spring and torque is measured.	50
4.4	The 3D-printed solid rods in Experiment 1 and three varied test conditions: infill density, infill pattern, and print orientation.	51
4.5	Experiment 1 results showing that E and G increase with infill density as well as more robust infill patterns. Tensile strength increases as printing angle increases; however, shear stress is highest at 45°	51
4.6	Experiment 2 results showing that 3D-printed helical springs perform similarly to theoretical predictions as measured by a load frame with different d , D , N , and L values.	53
4.7	Experiment 3 results showing that 3D-printed helical springs have similar twisting performance to theoretical predictions with varied d , D , N , and L values.	53
4.8	A prismatic joint is used for a linear-only deformation.	55
4.9	A revolute joint is used for a twist-only deformation.	55
4.10	A chain of knuckle joints used for bend-only deformations.	56

4.11	(a) A cylindrical joint is used for linear+twist deformations and (b) a chain of ball joints is used for twist+bend deformations.	57
4.12	The <i>Ondulé</i> spring design tool interface (left) has four parts: Rhino modeling environment, a spring generation panel, a spring stiffness control panel, and a spring behavior design panel. The workflow for each design panel is shown on the right.	58
4.13	Generating the medial axis, calculating the size of the selected body, and evaluating the printability of an embedded spring.	60
4.14	Generating the deformation spring using the generated medial axis and RhinoCommon functions.	61
4.15	Generating the decorative spring.	61
4.16	An exploded view of (a) a prismatic joint and (b) a chain of ball joints, which are used for linear-only and twist+bend respectively.	62
4.17	A jack-in-the-box spring mechanism generated by <i>Ondulé</i> : (a) two spring designs are embedded; (b) the cat can be fully compressed and locked inside a laser-cut box; and (c) the cat pops out following a path as a surprise. . . .	63
4.18	A launching rocket application: (a) a rocket sits on top of a compressed “smoke” spring, which is locked by an external latch; (b) the user can launch the rocket by pulling the latch; and (c) the smoke is in its full extension. . . .	64
4.19	Replicated and custom hand exercisers: (a) an off-the-shelf hand exerciser (b) a replication with 3D-printed springs, (c) a custom design that includes springs with different stiffnesses and custom prismatic joints for compress-only behavior, and (d) a blowfish version.	65
4.20	The setup of a tangible storytelling prop.	66
4.21	Other applications that <i>Ondulé</i> can support: (a) an accessible cutting device, (b) an elephant mask with a bendable trunk, and (c) a snake body with multiple spring deformation behaviors.	67
5.1	We introduce <i>Kinergy</i> , an interactive design tool to rapidly create 3D-printable energy-powered motion. Above, we show a 3D-printed pull-back car created with <i>Kinergy</i> : the static car model is converted into a motion-enabled model with an auto-generated and embedded spring, a spring lock, and a set of gears (left). All the parts in the converted 3D car model are printed in place with a commercial 3D printer and the printed car is ready to move without post-print part assembly (right).	70
5.2	Motion types and <i>Kinergy</i> kinetic unit examples.	73
5.3	Compliant lock mechanisms used in kinetic units: (a) guided bars and compliant latch designs for the locking control of the helical spring and (b) ratchet gear and compliant latch design for the locking control of the spiral spring. . .	75

5.4	For non-instant motion types, kinetic units combine geartrains (pairs of bull and spur gears) and kinematic transmission mechanisms: (a) rack-and-pinions, (b) axels or revolute joints, (c) Scotch yokes, (d) crank-and-slotted-levers, and (e) Geneva drives. Helical springs engage with gears through (f) rack-and-pinions and (g) spiral springs are co-axial with gears for driving the geartrain.	76
5.5	The user interface of <i>Kinergy</i> design tool.	79
5.6	The unit orientation, translation axes and rotation axes in each kinetic unit, and end-effectors and stationary parts.	82
5.7	The unit orientation, translation axis and rotation axis in each kinetic unit. The end-effectors are highlighted in light orange and the stationary parts are marked in grey.	82
5.8	The pull-back car 3D model is represented by a graph for motion preview. . .	87
5.9	Three types of tolerance identified as problematic for one-shot printing: (a) the gap between two mating gears, (b) the gap between a gear and its nearby spacer, and (c) the gap between a gear and its residing shaft. The 3D model is sliced in (d) an optimized orientation and the slicing settings are curated to create clean part contours for intermating elements.	89
5.10	The pull-back cars created with <i>Kinergy</i> are printed with various 3D printers and printing technologies: (a) industrial-level FDM 3D printing, (b) PolyJet 3D printing, and (c) SLA 3D printing.	89
5.11	An auto-opening umbrella prototype created with the instant translation kinetic unit: (a) the rendered 3D model of the umbrella, (b) the printed and assembled umbrella in the locked stat, and (c) the opened umbrella after the runner is unlocked.	91
5.12	A catapult-like game controller is made to virtually projectile birds in an Angry Birds game: (a) the rendered 3D model of the game controller, (b) the printed game controller with external sensor and circuitry embedded, (c) flying the bird at a small angle, and (d) flying the bird at a bigger angle. . . .	91
5.13	A continuous translation kinetic unit embedded switch is attached to a cardboard trash bin: (a) the rendered 3D model of the switch and the trash bin, (b) the printed switch, (c) the closed trash bin with the switch attached and locked, and (d) opening the bin's lid by unlocking the switch.	92
5.14	Four 3D-printed pull-back cars with different embedded energy, speed, and traveling distance: (a) the rendered car models and the corresponding printed cars (yellow: baseline; green: more stored energy; blue: faster; orange: longer travel distance), (b) the comparison of cars with different embedded energy, (c) the comparison of cars with different motion speed, and (d) the comparison of cars with different traveling distance.	93

5.15	A human-operated handheld flashlight created with <i>Kinergy</i> and external electronics: (a) the rendered model of the handheld flashlight, (b) the flashlight head and body, and (c) the functioning flashlight with the embedded kinetic unit.	94
5.16	A battery-free Maneki Neko sculpture with an embedded oscillation kinetic unit: (a) the rendered model of the cat sculpture, (b) the printed sculpture, and (c) the waving arm driven by the embedded kinetic unit.	94
5.17	The user holds a 3D-printed cutter that uses an embedded reciprocation kinetic unit to move the blade back and forth repeatedly and cut the bread: (a) the rendered model of the cutter, (b) the printed cutter, and (c) the functioning cutter.	95
5.18	A 3D-printed zoetrope installation that is driven by an intermittent rotation kinetic unit embedded base box and animates a 3D walking cat: (a) the rendered model of the zoetrope, (b) the printed setup with programmable LED flashlights, and (c) the rotating zoetrope in motion.	96
6.1	Example applications made with <i>FlexHaptics</i> method: (a) a piano keyboard interface for touchscreen musical applications, (b) a VR controller attachment for bow shooting games, and (c) a joystick with a two-step button on the stick end.	100
6.2	(a) A <i>FlexHaptics</i> module supports a haptic effect among resistance, detent, and bounce, along a linear or rotary path. The left six modules afford within-plane path, the two additional modules are designed for bounce effect along an out-of-plane linear path. Gray parts are rigid; colored parts are compliant, and color changing from blue to green, and to red indicates increasing stress levels. (b) All the moduels can be mixed for complex haptic feedback and moving paths.	103
6.3	Linear resistance module. It comprises a flexure slidable along a linear track. Its force feedback is adjusted with beam length l , thickness h , and width b	104
6.4	Rotary resistance module. It consists of a flexure rotatable within a ring. Its force feedback is adjusted with beam radian a , radius r , thickness h , and width b	105
6.5	Detent modules. A linear or rotary detent module employs the same beam geometries as linear resistance module and adapts notches to contact surface. (a) As the beam moving across a notch, force feedback is determined by the notch and beam geometry. (b) We present four symmetrical notch signatures and force-displacement curves. Mixing a left and a right side of them generates another 12 detent profiles. Force feedback from a notch can be adjusted by scaling it along its width or depth direction.	106

6.6	Bounce modules. (a) A linear bounce module can be stretched or compressed, its stiffness can be adjusted by beam length l , thickness h , and width b , and unit number n . (b) A rotary bounce module can be rotated clockwise or counterclockwise, its stiffness can be adjusted by spiral radian a and wire thickness h and <i>width</i>	108
6.7	Bounce ortho-planar modules. (a) A straight-beam ortho-planar bounce module can be adjusted with beam length l , thickness h and width b . (b) A curve-beam ortho-planar bounce module can be adjusted with beam radius r , radian a , and beam thickness h and width b	109
6.8	Two mixing operators for <i>FlexHaptics</i> modules. (a) Mixing in parallel aligns modules with the same movement path and bonds the mobile and static parts together respectively, resulting in an interface with the same movement path and a compound haptic effect. (b) Mixing in series uses multiple modules with different movement paths, and bonds the static part of one module to the mobile part of another module, producing an interface with a complex movement path.	110
6.9	<i>FlexHaptics</i> user interface consists of a <i>FlexHaptics</i> tool panel and module preview in Rhinoceros environment.	112
6.10	Application examples: (a) a slider input interface for touchscreen painting applications, (b) a piano keyboard interface for touchscreen musical applications, (c) a VR controller attachment for bow shooting games, (d) a string-based wearable haptic device, (e) a tactile low vision timer, and (f) a joystick with a two-step button on the stick end.	115
7.1	Two past projects on interactive tangible interfaces for children to (a) learn about an old language and (b) build custom wearable devices.	126
7.2	Two wearable devices that provide tactile feedback on body for blind and low vision users to (a) navigate and (b) fetch objects.	127
7.3	Two past projects that focused on (a) tactile overlays for making graphics accessible on touchscreens and (b) tactile wearable controller for a VR game.	128
7.4	Two past projects that used machine learning to add interactivity to fabricated objects: (a) converting squeezing gestures into interactive user input using acoustic sensing and (b) classifying spring's deformation behaviors using inductive sensing.	129
B.1	Material properties (a) Young's modulus E and (b) shear modulus G . G can be derived using E and Poisson ratio (c).	149

LIST OF TABLES

Table Number	Page
1.1 Commonly used 3D printable kinematic mechanisms and actuation controls to achieve kinetic behaviors.	3
1.2 Dissertation organization.	11
3.1 Search terms related to kinetic mechanisms and kinetic behaviors and their frequencies in the final complete set.	30
3.2 Frequencies of 3D-printable kinetic designs across 11 Thingiverse-defined categories.	33
4.1 The conditions of spring parameters used in Experiment 2 and Experiment 3.	52
5.1 Parameters for energy control and motion characterization in the kinetic units.	85

LIST OF SYMBOLS AND ABBREVIATIONS

HCI	human-computer interaction
FDM	fused deposition modeling
SLA	stereolithography
SLS	selective laser sintering
PLA	polylactic acid
ABS	acrylonitrile butadiene styrene
TPLA	tough polylactic acid
PVA	polyvinyl alcohol
CAD	computer-aided design
GUI	graphical user interface
UI	user interface
RG	research goal
MTM	Making Things Move
CSV	comma-separated values
DC	dissertation contribution
TPU	thermoplastic polyurethanes
DOF	degree-of-freedom
SMA	smart material alloy
ASTM	American Society for Testing and Materials
MCF	mean curvature flow
KU	kinetic unit
Rhino	Rhinoceros 3D
DFS	depth-first searching
VR	virtual reality

POM	polyoxymethylene
FEA	finite element analysis
STL	standard triangle language
SVG	scalable vector graphics
MJF	multi jet fusion
PBF	powder bed fusion
SUS	system usability scale
MR	mixed reality
AR	augmented reality
VR	virtual reality
XR	extended reality

ACKNOWLEDGMENTS

This has been a long journey full of joys and tears. There are moments of accomplishments; there are moments of frustrations; there are moments of overwhelm. Yet, I feel deeply grateful that I am not alone on this journey. As I am arriving at the destination of this once lifetime, unforgettable journey, I would like to take this opportunity to thank all the people who have accompanied, supported, and helped me reach this significant milestone of my life.

First, I want to express my sincere gratitude to my family, especially my parents—Yan Lou and Jianhua He. Your selfless love and support allow me to live with my dream and chase my passion. As medical doctors, you show your kindness to the patients, which teaches me to be a kindhearted, attentive, empathic person in work and life. The most important lesson I learned from my parents is to have unrelenting persistence in doing every job.

Second, I would like to thank the person who is always inspirational to me and instrumental to my PhD study: Jon E. Froehlich. Thank you for being an amazingly supportive, patient, and talented advisor. I immensely appreciate the freedom and advice you gave me in the past few years, which enabled me to establish my unique research path towards becoming an independent HCI researcher. Your selfless support for your students, your passion for teaching, and your enthusiasm for community building show me the essential components to succeed in academia, which will have a lasting impact on my career. I also want to thank you for allowing me to be part of the Makeability Lab. I feel lucky to be surrounded by so many amazing labbers in the past seven years at the University of Maryland and the University of Washington. Thank you again for your instrumental guidance throughout my PhD study. And I am inspired to be an impactful researcher like you in my future career.

I feel fortunate and grateful to spend the past seven years of my life with a group

of amazing people at the Makeability Lab. First, I would like to thank Kotaro Hara and Matthew Mauriello. Thank you for your guidance and help during my PhD's first few years. Your diligence and determination in research and academia always inspire me to continue my career in academia. I am honored to follow your path to become an Assistant Professor and will continue looking up to you guys. Lee Stearns and Seokbin Kang, thank you for always being in the lab and bringing the maker spirit into the lab. I respect your hard work and your pursuit of perfection. Manaswi Saha, thank you for being my academic sibling and accompanying me from the beginning of this journey. I will remember the moments that saw our growth together and celebrated our success over the past seven years. Best of luck with your job searching. Dhruv Jain, you are a fantastic friend who I can always count on. Thank you for listening to my complaints and helping me go through my professional and personal life's ups and downs. I cannot wait to see the excellent work you will produce as a faculty member. I also want to thank all the younger Makeability Lab members, including Mikey Saugstad, Jesse Martinez, Daniel Campos Zamora, Xia, Su, and the two incoming students—Chu Li and Jaewook Lee. Thank you for your friendships. It is so wonderful to have overlappings with all of you. Mikey, I always love your sense of humor, and thank you for bringing vibrant morale to our lab. Jesse, you are creative, and I respect and admire your determination to accomplish something you are really into. Daniel and Xia, it was my honor and pleasure to work with you and learn from you. Those days of tinkering and fabrication in the lab will be unforgettable in my life. Chu Li and Jaewook Lee, thank you for your support and energy, and I believe you will thrive in the Makeability Lab with such a group of beautiful people.

My gratitude also goes to my excellent academic and industrial collaborators, who made my PhD truly special and helped me achieve academic goals: Jeffery Lipton, Huaishu Peng, François Guimbretière, Mark Fuge, Hongnan Lin, Tingyu Cheng, Hyun Joo Oh, Rafael 'Tico' Ballagas, Kris Erickson, Jarrid Wittkopf, Ji Won Jun, Rob DeLine, Saleema Amershi, Gierad Laput, Eric Brockmeyer, Majeed Kazemitabaar, Beryl Plimmer, Kasun Karunanayaka, Kelvi Cheng, Ellen Yi-Luen Do, Xuhai Xu, Venkatesh Potluri, Jennifer

Mankoff, Zijian Wan, Leah Findlater, Danli Wang. Finally, I also would like to express my gratitude to Microsoft Research and HP Labs for giving me the opportunities to gain research experience in the industry.

Finally, I want to thank many other faculty members and friends who always generously supported my PhD study, including Adriana Schulz, Nadya Peek, Stefanie Mueller, Ryo Suzuki, Yang Zhang, Pedro Lopes, Jeeun Kim, Valkyrie Savage, Toby Jia-Jun Li, Ken Nakagaki, Ge Gao, Diyi Yang, Teng Han, Yuhang Zhao, Anhong Guo, Xu Wang, Haijun Xia, Robert Xiao, Edward Wang, Cindy Hsin-Liu Kao, Scott Hudson, Ali Momeni, Thijs Roumen, Jun Nishida, Tongshuang Wu, Haojian Jin, Xing-Dong Yang, Ruofei Du, Junyi Zhu, Yueqian Zhang, Jessica Chin, Michelle Lin, Yuebing Liang, Yawen Zheng, Joshua Land, Sophie Tian, Xiyuan Shen, Arjun Simha, Mingrui Zhang, Qisheng Li, Xingfan Huang, Wenjun Wu, Hanchuan Li, Xiaoyi Zhang, Jasper O’Leary, Danli Luo, Dingzeyu Li, Jun Gong, Qiuyue Xue, Haichen Shen, Danyang Zhuo, Zhen Zhang, Chunjong Park, Xin Liu, Xuan Luo, Beibin Lin, Yang Liu, Eunice Jun, Christine Chen, Pratysh Patel, Chandrakana Nandi, Vikram Iyer, Lisa Orii, Katherine Juarez, Kelly Mack, Ather Sharif, Jonggi Hong, Uran Oh, Meethu Malu, Joche Choi, Karthik Badam, Hanzi ‘Lotus’ Zhang, Junhan ‘Judy’ Kong, Mingyuan Zhong, Ding Xu, Cheng Xu, Mei Xue, Zhen Geng, Mengying ‘Cathy’ Fang, Meng Shi, Weize Kong, Yunkai Zhang, Chao Wu, Jinjin Ge, Ze Jin, Qingqing Fan, Tao Liu, Xieyang Liu, Mingzhe Li, Yan Chen, Zhuoshu Li, Qian Yang, Wei Fu, and Hanxi Zhou, Jingyi Li, and Yunyi Zhu.

DEDICATION

To my advisor, friends, and family, I couldn't have done this without you.

Thank you for all of your support along the way.

Chapter 1

INTRODUCTION

3D printing, as an emerging prototyping technology, promises to enable designers and makers to rapidly create physical objects [7], from a rigid Stanford bunny to complexly textured decors. The past decades have seen an increase in sales for the 3D printer market (Figure 1.1). As a result, people have easier access to affordable desktop 3D printers at home, and traditional large-scale manufacturing has shifted toward personal-scale digital fabrication [7]. Meanwhile, home 3D printers have been developed to support various 3D printing methods, such as Fused Deposition Modeling (FDM) and Stereolithography (SLA), and materials like rigid plastics and elastic resins. All these technological advances enable a wide variety of applications (Figure 1.1). For example, a Master Yoda figurine¹ can be printed on a ~\$400 FDM desktop 3D printer using PLA materials, while precise dental splints can be printed on an SLA 3D printer using a biocompatible material—Dental LT Clear Resin.² However, while 3D printing technology has long been touted as a technique to revolutionize manufacturing, transform rapid prototyping, and enable personalized fabrication, most 3D-printed objects created with home 3D printing are still made with static forms [1, 141, 8] and lack interactivity, which limits the potential of 3D printing technology and the creativity of 3D model makers.

Converting static 3D models into non-static and interactive objects, however, is inherently difficult and introduces unique challenges in Human-Computer Interaction (HCI) [21, 18]. First, there are high barriers to making kinetic objects. For example, to make a functional watch, the designer needs to have knowledge of the underlying mechanical principles. Prior experience with fabrication, such as 3D printing or laser cutting, can also help with the design decisions. When the making involves manual assembling, additional

¹Master Yoda sculpture: <https://www.thingiverse.com/thing:4038181>

²3D printing dental splints: <https://dental.formlabs.com/uk/indications/splints-and-occlusal-guards/guide/>

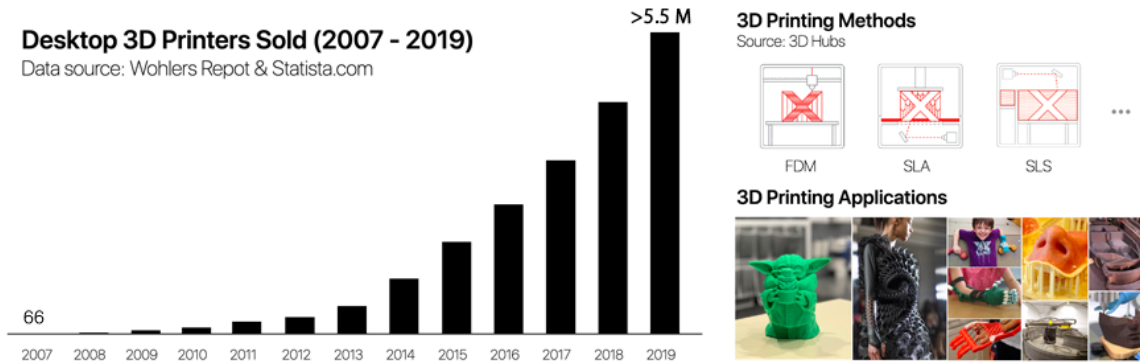


Figure 1.1: There is an increase on the sale of 3D printers from 2007 to 2019 before the pandemic (left). The common 3D printing methods include FDM, SLA, SLS, *etc.* and 3D printing applications benefit from the advances of 3D printing technology and 3D printable materials (right).

crafting skills are also needed. Second, 3D models of kinetic objects are not straightforward with existing Computer-Aided Design (CAD) tools. CAD tools are made for 3D modelers to create custom rigid components, such as individual machinery parts (*e.g.*, Solidworks³) and organic 3D shapes (*e.g.*, Rhinoceros 3D⁴). There is a high learning curve for end-users, especially novices, to master advanced operations with the complex CAD interface in order to design and parameterize kinetic structures precisely (not all CAD tools support the parametric design of 3D models). Finally, assembly is usually cumbersome and error-prone for those complex models that need combinations of multiple parts. For example, creating a functional 3D printable pull-back car model requires trial-and-error to determine the right tolerance for inter-engaged gears and a strict sequence for a complete and successful assembly.

Recent work has explored making movable 3D printable objects by embedding 3D printable mechanically functional components, such as joints [16], hinges [89], and metamaterials [41]; these components are traditionally used in machinery and material sciences. With the capabilities of 3D printers, these mechanical components can be modeled, printed, and em-

³Solidworks: <https://www.solidworks.com/>

⁴Rhinoceros 3D: <https://www.rhino3d.com/>

Table 1.1: Commonly used 3D printable kinematic mechanisms and actuation controls to achieve kinetic behaviors.

		Primary 3D Printable Kinetic Mechanism				
		Joint & Hinge	Microstructure	Linkage	Gear	Others (crank, cam, telescoping, etc.)
Actuation Type	Manual Control	Articulate [13, 16], rotate & move [121], transform [79, 91, 136, 142], compress, stretch, & bend [131]	Shear [41, 44], controllable deformation [10, 83, 104], articulate [110]	Rotate [14, 48]	Rotate & oscillate [38, 92, 138, 129]	Controllable deformation [134]
	Environmental Stimuli	Bend [73, 89, 94] controllable deformation [45] compress & stretch [68]	Transform [4], move & rotate [82], controllable deformation [26, 27, 29, 54, 124, 125]			
	External Actuator	Controllable movement [71], bend [90]	Controllable deformation [20], controllable movement [55]	Controllable movement [61, 64, 128], move [62], rotate [72]	Controllable movement [77]	Rotate [108, 109]

bedded into 3D objects to achieve kinematic movements, such as articulation [13, 16] and rotation [72], which are otherwise impossible with static 3D models.

Another research direction uses environmentally responsive materials such as foams [50] and thermoplastics [124]. Most of these approaches are used for creating shape-changing interfaces under strict control of environmental stimuli [26, 124, 27, 89]. Finally, researchers have also explored novel fabrication machines and methods to enable fabricated devices with kinematic properties [40, 86, 85]. These approaches require sophisticated fabrication processes or custom fabrication machines to achieve desired behaviors. For example, a custom printer head was designed and created for wool yarn feeding and printing wool-based elastic objects [40].

Table 1.1 summarizes a few commonly used 3D printable kinematic mechanisms and actuation controls for kinetic behaviors. Building upon prior work, I found that 3D printable springs have not been extensively studied for 3D printing but present interesting potential to make 3D printable kinetic objects [23]. Therefore, my dissertation aims to answer two research questions: *What kinetic objects can be created using 3D printable springs for new interactive applications?* and *How can we enable end-users to design and control these 3D printable spring-based objects?*

To answer these questions, I first investigate how makers create kinetic 3D models in general, focusing on how springs are used in 3D printing through a large-scale study on

Thingiverse. Then, I identify three behaviors that 3D printable springs can support for interactions: deformation, actuation, and I/O functions, such as haptic feedback and sensing. I propose spring-based kinetic mechanisms that are parametric and can be embedded in fabricated objects for desired kinetic behaviors. Finally, by controlling the mechanical and material properties of these spring-based mechanisms, I demonstrate the potential of this approach for adding deformation, actuation, and sensing capabilities to custom fabricated objects, which also informs a new spectrum of personal fabrication applications (Figure 1.2).

1.1 Mechanical Springs for 3D Printing

In contrast to other kinetic mechanisms explored for 3D printing, mechanical springs offer a variety of unique benefits as the primary mechanisms studied and used in my dissertation. First, they support various parameterizable deformation behaviors (*e.g.*, compress, stretch, bend, and twist) [11]. Second, they are ideal to store energy when deformed and produce the driving force by releasing the stored energy [11]. Finally, they offer versatile functions when combined with other mechanical components, such as gears [108] and levers [117]. The compelling potential of mechanical springs opens up new opportunities for end-users to create non-static 3D printable objects for interactions. Unlike off-the-shelf steel springs usually manufactured on demand, 3D-printed springs allow users to freely customize, rapidly fabricate, and easily embed desired spring forms into arbitrary 3D models.

While researchers have reviewed some spring forms for 3D printing and laser cutting, such as leaf springs [9], planar flexures [59, 140], and spiral springs [46, 113, 117], three key challenges prevent them from being widely used in FDM 3D printing. First, due to the anisotropic characteristics of FDM 3D printing, the performance and mechanical properties of 3D-printed springs are primarily influenced by the printing orientation. Second, since the mainstream material supported by consumer-grade 3D printers is plastic, 3D-printed springs are less durable, and the print quality mainly depends on the 3D printing methods. Finally, although springs support expressive deformations, designing, customizing, and controlling the deformation is complex.

In my dissertation, I first investigate how spring performance is affected by varied print-

ing settings (*e.g.*, printing orientation, infill density) and explore application scenarios where plastic 3D-printed springs are appropriate for desired functions. Further, I develop custom interactive design tools that enable 3D modelers, who have 3D modeling experience but not necessarily mechanical background, to design, create, and control 3D printable kinetic objects using parametric in-place spring-based mechanisms, which can be fabricated on consumer-grade 3D printers and reduce human work for manual assembly.

1.2 Research Approach

In my dissertation, I take a cross-disciplinary approach to develop design techniques and tools that increase the expressivity of consumer-grade 3D printing beyond static and rigid shapes.

First, I reviewed literature and theories to understand the principles of mechanical springs, identifying parameters that influence the spring's mechanical performance, such as tension and torsion behaviors. These exposed me to a substantial amount of concepts, terminologies, and principles in mechanical engineering. In addition to spring theories, I also studied the principles for the functioning of commonly used kinematic components, such as gears and joints. Those components play an essential role in working with the core structures—spring-based mechanisms—for various kinetic behaviors, *i.e.*, deformation, motion, and sensing.

Then, I followed an iterative design and prototyping process to create unique spring-based mechanisms suited to achieve desired kinetic behaviors in various 3D printing applications. Informed by the standard spring designs from the theory, I devised unique spring-based mechanism designs by combining parametric spring forms and kinematic components if needed for specific behaviors, such as deformation and actuation. To derive these designs, I followed an iterative trial-and-error process, including 3D modeling and evaluations through printing and prototyping. The deep understanding of mechanical springs and kinematic components principles and the empirical experiments and prototypes enabled me to develop further parametric, customizable, controllable design primitives for 3D printing and novel applications in this dissertation.

In addition to design techniques, I also built interactive design tools for end-users using

principles and skills in HCI. These tools, made as plugins to Rhinoceros 3D, prepare the proposed spring-based mechanisms as templates, which are parameterizable by the end-user through graphical user interface controls, and automate the integration of these novel spring-based mechanisms into custom 3D models. In general, the workflow using these tools is for the end-user to (i) create and edit 3D models in the default 3D modeling scene in Rhino, (ii) select and embed a proposed spring-based primitive into the custom 3D model, (iii) customize the embedded behavior enabled by the mechanism via interactive user controls (*e.g.*, buttons and sliders) in the user interface, (iv) preview the auto-generated 3D design that converts high-level behavior characteristics input by the user into underlying geometries of spring-based mechanisms, and (v) export 3D printable files for fabrication with home 3D printers.

For evaluation, I demonstrated the potential of the proposed techniques and tools via a series of functional application prototypes. This dissertation presents new design opportunities for developing kinetic 3D objects with spring-based mechanisms. Its results should also benefit researchers working to add interactivity to fabricated objects in the fields of HCI, mechanical engineering, and robotics.

1.3 Dissertation Goals and Contributions

The research goal of this dissertation has been to design, develop, and evaluate 3D printable spring-based mechanisms to promote the fabrication of kinetic objects for interactivity. My approach was threefold: (i) to draw upon mechanical engineering, physics, and practices of making kinetic 3D designs within the maker community to inform the use of mechanical springs in creating kinetic objects and supporting interaction; (ii) to design and evaluate parametric spring-based mechanisms that utilize mechanical springs and kinematic elements for desired kinetic behaviors, including deformation, actuation, and sensing through structural form changes; and (iii) to develop and evaluate interactive parametric design tools for end-users to create custom 3D printable kinetic models with embedded spring-based mechanisms for a variety of applications.

Dissertation Statement: We can design 3D modeling and printing techniques to embed and control parametric spring-based mechanisms into 3D-printable

objects, which enables a new suite of applications for 3D printing.

1.3.1 Understanding the Role of Springs in Creating 3D Printable Kinetic Designs

The first goal of this dissertation was to define how springs could be potentially used to support the creation of 3D printable kinetic objects. To achieve this goal, I studied two areas: (i) spring theory to understand how springs are formed and performed in machinery, and (ii) practices in making 3D printable kinetic designs to identify if springs are possible to support in this field. For the former, I wanted to be able to map spring parameters to certain spring behaviors and extend the fundamental principles [11] of making mechanical springs for 3D printing. My research validated the mechanical performance of 3D printable springs through a series of experiments. The results informed the design of spring-based mechanisms for concrete behaviors, including predictable deformations, force feedback, actions, and sensing capabilities. Chapters 4-6 describe how various forms of mechanical springs were examined and adapted for 3D printing to achieve desired kinetic behaviors.

For the second area, I aimed to unveil how makers are making 3D printable kinetic objects, characterize what mechanisms, tools, techniques, and types of kinetic objects are being made, and identify the trends, challenges, and opportunities for making 3D printable kinetic objects. Compared with the past fabrication research, I also wanted to identify the gap between the fabrication research and the maker community and explore how spring-based approaches could bridge this gap. To capture all these aspects, I conducted a *Making Things Move (MTM)* study, which examines how makers create kinetic 3D printable models on Thingiverse, and created an open-source dataset of representative kinetic designs on *Thingiverse*. Chapter 3 describes the dataset, the detailed study methods, findings, and insights that inform how springs could be used to enable new design and fabrication of kinetic objects.

1.3.2 Developing Spring-Based Mechanisms for Kinetic Behaviors

The second goal of this dissertation was to develop novel 3D printable spring-based mechanisms to achieve identified interactive functions. These functions should cover both inter-

active input and output behaviors. For example, how could we add sensing capabilities to kinetic 3D models to interpret user input for interaction? How could we create controllable deformation or movement for kinetic 3D objects? My research explores the parametric design space of multiple forms of springs to achieve desired I/O functions. To demonstrate that springs can be used to support diverse input and output behaviors, I developed a set of helical spring and joint based design techniques that are embedded into 3D models for controllable deformation behaviors in *Ondué* (published at UIST'19 [32]), a series of parametric units that combine helical or sprial springs with kinematic transmission mechanisms to transfer potential energy into desired output motions in *Kinergy* (to appear at UIST'22 [33]), and a suite of compliant flat beam structures to construct custom haptic input interfaces in *FlexHaptics* (published at CHI'22 [65]). In these systems, all the spring-based mechanisms were designed to support 3D printing in place so that they could be printed together with the 3D models in one shot without further manual assembly.

1.3.3 Building and Evaluating Tools for Creating Spring-Based Kinetic Objects

With the spring-based mechanisms described above, the third and final goal was to lower the barrier for end-users to design and control 3D printable kinetic behaviors using the proposed spring-based mechanisms. Informed and motivated by similar design editors [41, 134], I developed a series of interactive parametric design tools for the end-user to embed spring-based mechanisms, parameterize the mechanisms for desired output behaviors, and preview the results in a 3D environment. These tools are targeted at 3D modelers who are proficient in modeling and design but lack knowledge in mechanical engineering to create mechanically kinetic designs.

While distinct spring-based mechanisms were applied for different purposes, these custom tools share three commonalities. First, the end-user can embed one of the provided in-place kinetic mechanisms in the 3D shape for desired behavior. Second, the user can parameterize the behavior by editing the spring-based mechanism directly in the body. Finally, the user can validate the 3D printable behavior via a preview. To achieve these functionalities, I extracted the underlying mechanism parameters and associated them with

high-level user requirements, which could be input and controlled by the end-user through the user interfaces. Geometric constraints and printability of the embedded in-place mechanisms were also considered when the spring-based mechanisms were integrated into the 3D object. As a result, the tools make it easy for the end-user to understand the function of the provided spring-based mechanisms, follow instructional steps for embedding spring-based mechanisms into 3D models, and validate resulting designs via real-time updates in 3D models.

To evaluate these enabling tools, I validated the feasibility of the proposed techniques by making 3D printable kinetic, functional objects with the tools. As a result, I created prototypes across different application domains and demonstrated the kinetic functions achieved by these prototypes. Chapters 4-6 present a variety of applications to showcase how different spring-based mechanisms are used to create deformable objects, desired self-propelled motions, custom haptic interfaces, and predictable sensing functions.

1.4 Summary of Contributions

In summary, this dissertation contributes: (i) *a large-scale analysis* that studies current trends in making 3D printable kinetic objects and informs challenges and opportunities for using springs in 3D printable movable objects through an open-source dataset of representative kinetic designs on *Thingiverse*; (ii) *a set of parametric designs of spring-based mechanisms* for desired kinetic behaviors such as deformation, actuation, and sensing; (iii) *custom interactive design tools* that allow 3D modelers to design and control 3D printable kinetic behaviors using in-place spring-based mechanisms; and (iv) *a series of proof-of-concept applications* enabled by proposed design techniques and tools.

My dissertation unveils the potential of 3D printable springs for creating kinetic 3D-printed objects. While prior work has explored converting static 3D-printed objects into deformable shapes, actuators, and sensors via different approaches, this body of work focuses on using 3D printable spring-based mechanisms to enable all three behaviors (Figure 1.2).

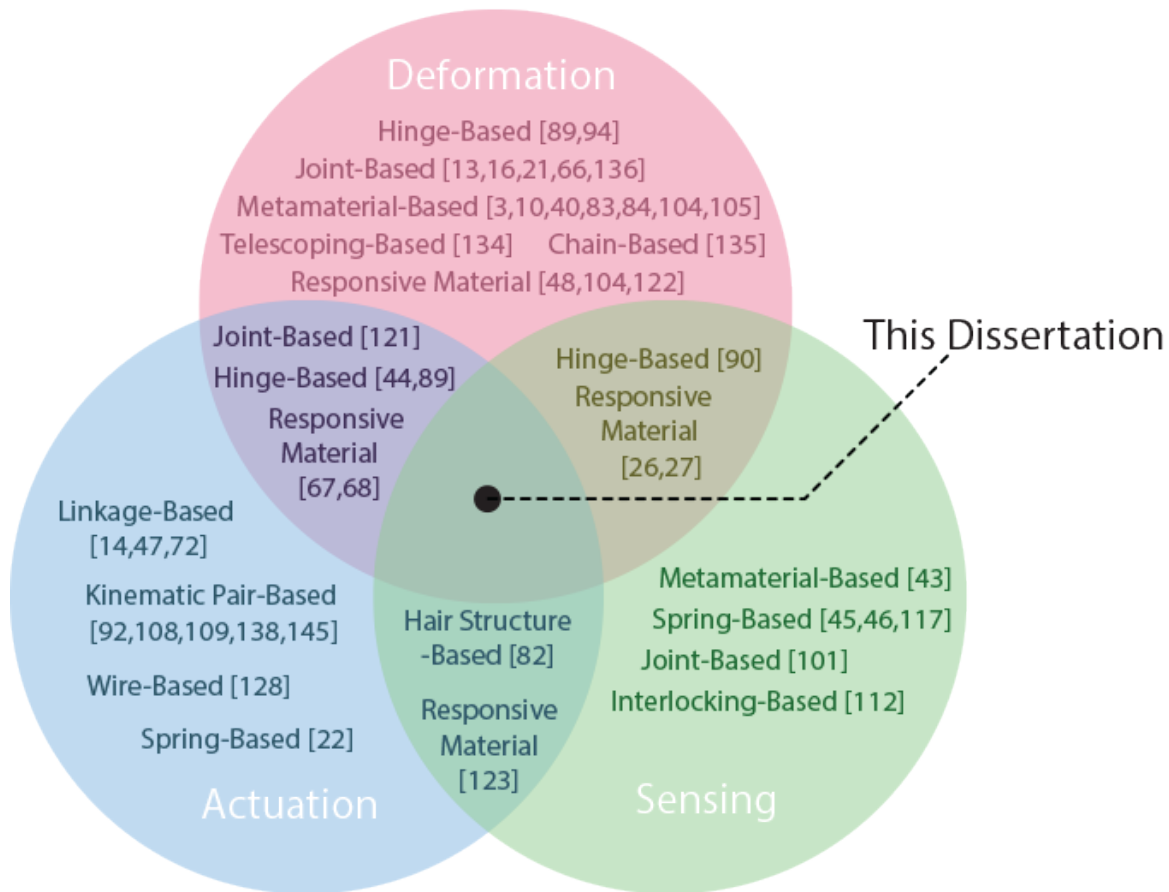


Figure 1.2: This dissertation focuses on how to use 3D printable spring-based mechanisms to enable deformation, actuation, and sensing behaviors.

1.5 Dissertation Organization

The rest of this dissertation is structured as follows. Chapter 2 briefly reviews related work and background, and Chapter 3 presents a study analysis—*Making Things Move (MTM)*—on how makers create 3D kinetic designs. Chapter 4 describes project *Ondulé*, which embeds 3D printable helical springs into 3D models for controllable deformation behaviors; Chapter 5 presents project *Kinergy*, which enables 3D printable objects to move by translating the energy stored in the embedded spring into desired output motion via specialized kinematic components; and Chapter 6 discusses *FlexHaptics*, which was led by Dr. Hongnan Lin

at Georgia Institute of Technology and explores how to utilize various flat beam spring forms to create interfaces with custom haptic feedback and even add sensing capabilities to structurally deformable fabricated objects. Table 1.2 shows the contributions of each chapter and how each chapter is organized in response to addressing the research questions. Finally, the dissertation concludes in Chapter 7 and describes the future work to be done after this dissertation.

Table 1.2: Dissertation organization.

Research Questions	Contributions	Addressed In
RQ1	DC1	Chapter 3: Through the study of <i>MTM</i> —a study on how makers create 3D printable kinetic objects
RQ1, RQ2	DC2, DC3, DC4	Chapter 4: Through the implementation and evaluation of <i>Ondulé</i> —designing and controlling 3D printable deformation behaviors with embedded helical springs
RQ1, RQ2	DC2, DC3, DC4	Chapter 5: Through the implementation and evaluation of <i>Kinergy</i> —creating custom objects with 3D printable self-propelled motion
RQ1, RQ2	DC2, DC3, DC4	Chapter 6: Through the implementation and evaluation of <i>FlexHaptics</i> —creating custom haptic input interfaces with planar compliant structures

RQ1: What kinetic objects can be created using 3D printable springs for new interactive applications?

RQ2: How can we enable end-users to design and control these spring-based objects?

DC1: A large-scale analysis of an open-source dataset of kinetic models on *Thingiverse*, which investigates and characterizes the current trends on *Thingiverse* to reveal challenges and opportunities for making 3D printable kinetic objects.

DC2: A set of parametric spring-based mechanisms for desired kinetic behaviors such as deformation, actuation, and sensing.

DC3: Interactive design tools that allow 3D modelers to design and control 3D printable kinetic behaviors using in-place spring-based mechanisms.

DC4: A series of functional applications created with the proposed techniques and tools.

Chapter 2

BACKGROUND AND RELATED WORK

This dissertation presents a suite of parametric spring-based mechanisms and tools that support the design and control of 3D printable kinetic behaviors for interactivity, focusing on controllable deformation, actuated motion, and sensing capabilities. To inform those novel spring-based mechanisms and develop user-friendly user interfaces, I situate my work within the intersection of multiple fields, including mechanical engineering, spring physics, and HCI. This chapter summarizes the status quo fabrication research on making kinetic objects for input and output functions.

In this chapter, I begin by drawing upon the physics theory of mechanical springs and providing an overview of the parametric design of various spring forms for 3D printing (Section 2.1). I then review related work on three main threads: making 3D printable objects with deformation behaviors (Section 2.2), actuating 3D-printed objects for interaction (Section 2.3), and adding sensing capabilities to 3D printable objects (Section 2.4). In addition, my dissertation also relates to the large group of parametric tools for designing and controlling 3D printable functions. Consequently, I finally review the literature on developing computational tools that enable end-users to embed, edit, and simulate custom design primitives to 3D models for desired output behaviors (Section 2.5).

2.1 Spring Physics and Theory

Springs are elastic structures that can harness mechanical energy and support a wide range of deformation behaviors. In general, springs are classified as wire springs (*e.g.*, helical springs), flat springs (*e.g.*, cantilever), and special-shaped springs [11]. Regardless of the spring forms, a spring exerts an opposing force approximately proportional to its deflection change when it is deformed from its resting position. The deformed spring stops exerting any energy until it reaches its equilibrium. This unique spring characteristic enables springs

to exhibit a diverse range of deformation behaviors and distinct spring forms usually are manufactured for particular dominant deformation purposes [11]. For example, although tension springs and compression springs are all made with a series of coils, the turns in a tension spring usually touch in the unloaded position while not in the unloaded position in a compression spring. A spiral-based flat spring is easier to wind than perform ortho-planar stretching. In addition, deformed springs have potential energy stored in their deformed forms. When springs start restoring their equilibrium state, the stored potential energy converts into kinetic energy, resulting in spring motion [11, 44]. As a result, springs have the potential to work as energy motors. This dissertation explored embedding deformable spring shapes into 3D printable objects to enable the change of 3D shapes and imbuing 3D printable springs as self-contained motors into 3D models for producing output motion.

Besides the benefits offered by springs, spring theory also informs the design and manufacturing of different spring types and how the spring parameters influence the mechanical performance of springs [11, 2]. For example, a helical spring’s compression and extension behaviors are determined by the spring diameter D , wire diameter d , and the number of coils N (Figure 2.1a). As the spring diameter increases with thinner and closer coils, the spring becomes more elastic and soft. In a spiral spring, thicker and fewer coils make the spring harder to wind. The torque a spiral spring can produce depends on the spring coil thickness t , the number of coils N , and coil width b (Figure 2.1b). A beam spring (cantilever) can bear different loads by varying beam length l , beam thickness t , and beam width b (Figure 2.1c). The relationship between the set of spring parameters and the mechanical behavior of the spring provides guidance for me to test, design, control, and fabricate 3D printable springs for desired behaviors. In this dissertation, I use 3D printable helical springs, spiral springs, and beam springs as the core mechanisms and imbue 3D objects with these parameterizable springs to achieve deformation behaviors, haptic feedback, actuation, and sensing.

2.2 Making 3D Printable Objects with Deformation Behaviors

Past work introduced techniques to make movable 3D-printed objects with deformation behaviors for different purposes via mechanical elements such as joints [13, 16, 136] and hinges [45, 89] or via responsive materials and structures that react to environmental stimuli

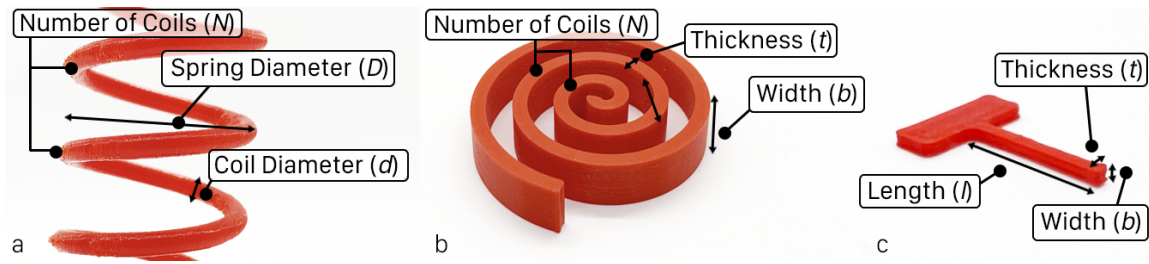


Figure 2.1: Three common spring types and parameters: (a) helical spring, (b) spiral spring, and (c) beam spring.

such as heat [50, 94, 124] and pressure [68, 131, 95], *e.g.*, creating articulated characters [13, 16], transformable objects [135, 142], and shape-changing interfaces [27, 124]. Below, I discuss the relevance of these approaches to the work for my proposal.

In traditional mechanical assemblies, mechanical parts such as gears, cams, and cranks are interconnected and move to achieve specific mechanical and kinematic functions such as translating and force transmission. Researchers in HCI’s digital fabrication community have also explored using mechanical elements to create articulated and reconfigurable 3D-printed objects.

Joints. Mechanical joints are a class of machines that connect one or more mechanical parts to another and are designed to allow relative movements of those mechanical parts. There are many types of joints, such as prismatic joint, ball joint, revolute joint, and knuckle joint. Prior work investigated how to 3D print functional joints for mechanical behaviors [21, 38, 66]. For example, Calì *et al.* [16] converted static 3D models into articulated ones using 3D printed ball joints (Figure 2.2a). By controlling the friction between the ball and the socket in the joint design, the proposed method can generate printable and assembly-free ball joints that allow the model to hold a pose. Similarly, Bächer *et al.* [13] introduced a way to convert an input geometry into articulated deformable characters by placing friction joints on the medial axis representation of the input geometry (Figure 2.2b). In addition to making articulated characters, joints are also used for reconfiguring the shape of 3D-printed objects [142]. For example, Yuan *et al.* [136] developed a computational approach to transforming common object shapes into target character forms by embedding joints (Figure

2.2c). The approach follows a set of guides in an iterative process to model the attributes of desired transformable designs and optimizes the embedded joints to approximate the resulting character configuration.

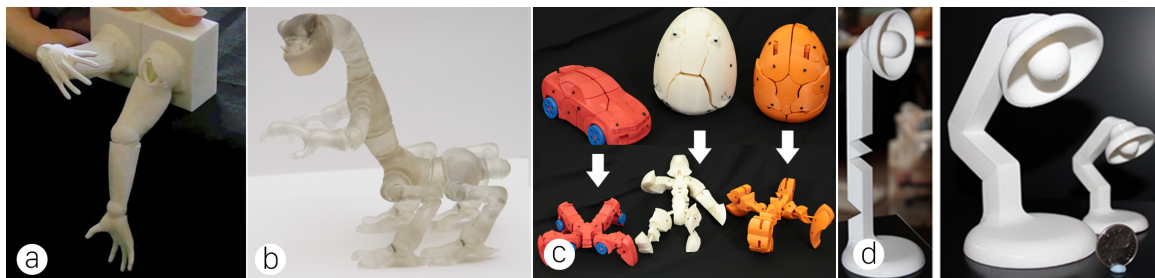


Figure 2.2: Articulated and transformable objects made with embedded 3D printable joints and hinges.

Hinges. To make bendable, twistable, and foldable objects, 3D printable hinges are embedded in 3D models [45, 89, 90]. For example, *Meltables* [94] is a novel fabrication approach to creating complex 3D shapes by embedding a set of planar beams as hinges in 3D-printed objects and deforming those hinges when heated for target shapes (Figure 2.2d). *Pop-up Print* [79] uses uniquely designed hinge mechanisms to achieve folding/unfolding behaviors for 3D-printed bodies. The object can be printed in a compact “folded” state and then unfold to the target shape due to the reversibility of folding/unfolding, saving printing volume and time (Figure 2.3a).

Metamaterials. Recently, another group of work enables 3D-printed augmented behaviors such as deformation and transformation using mechanical metamaterials [3, 10, 42, 44, 70, 84, 105]. *Metamaterials* [43, 41] are repeating cell structures embedded in base materials with varied shapes, sizes, orientations, and arrangements for advanced material and mechanical properties. Within HCI, integrating metamaterial designs in 3D-printed objects can produce enhanced object behaviors. For example, Ion *et al.* [41] introduced the concept of metamaterials to digital fabrication in HCI by employing a block of 3D printed shear cells to control directional movements (Figure 2.3b). The tessellated cells in *KinetiX* [83] perform shearing and bending under manual compression (Figure 2.3c). Finally, Schumacher

et al. [104] developed a data-driven method to tile the 3D object’s interior with small-scale microstructures to achieve desired material properties. By controlling the microstructures, objects printed with stiff material can have the effect of a softer material in selective 3D regions (Figure 2.3d).

Other Mechanical Elements. To achieve desired functions, other mechanical elements and structures are also used for 3D-printed objects [92, 112, 134]. For example, Yu *et al.* [134] introduced a tool to convert 3D models into telescoping designs that can resize the object body along with the skeleton (Figure 2.3e). In addition, *LineUp* [135] converts 3D models into chain-based structures and then transforms the shape between different configurations of the 3D printable chain structures (Figure 2.3f).

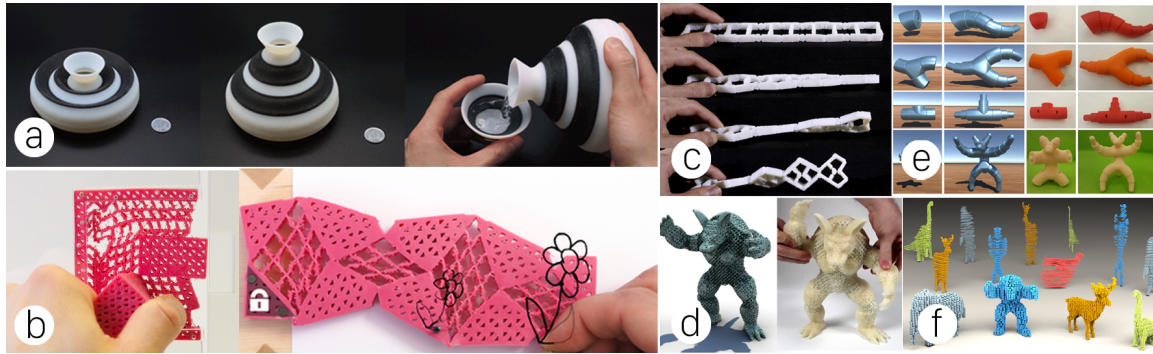


Figure 2.3: Deformable 3D-printed objects created with hinges, metamaterials, and other mechanical elements.

While the mechanical performance of 3D printable springs has been studied recently [9, 81, 93, 100], 3D printable mechanical springs have the potential to create deformable objects. Furthermore, compared with other mechanisms such as joints and hinges, springs can be parameterized for controllable behaviors such as compressing, bending, and twisting [17]. As part of my dissertation, 3D printable springs are studied and used to support a variety of deformation behaviors that can be easily integrated into 3D-printed objects.

Besides mechanical elements, researchers have also explored creating deformable objects with non-plastic materials such as silicone and elastic resin that can modify the material properties of the 3D-printed objects, producing versatile behaviors. For example, Zehn-

der *et al.* [137] injected spherical inclusions of a liquid dopant material into a silicone matrix material. They controlled the composite silicone rubber’s macroscopic mechanical properties by varying the injected inclusions’ number, size, and locations (Figure 2.4a). In addition, Skouras *et al.* [106] introduced a method to compute and optimize the locations and distribution of elastic material inside a 3D printable object for target poses (Figure 2.4b).

Another large body of research explores creating shape-changing interfaces with embedded material and structure composites sensitive to heat [27, 50, 116]. When these embedded composites are under heating, their intrinsic material and structural properties mutate, resulting in the macro changes in the shape. Desired shape-changing behaviors such as translating, folding, and scaling can be achieved by designing and parameterizing these composites in 3D-printed objects. For example, *Geodesy+* [29] is an end-to-end fabrication tool that generates optimized tool paths for extruding thermoplastic and transforms a 2D flat sheet into a 3D geometry in hot water (Figure 2.4c). Similarly, in hot water, *A-line* [124] triggers a single straight line to self-fold into target 2D or 3D wireframe structures with unique linear arrangements of thermoplastic material composites in the 3D-printed line structure (Figure 2.4d). By embedding precise heating structures along with moldable and non-modified regions in 3D-printed objects, *HotFlex* [26] turns the 3D-printed objects into the deformable state under heating and enables the user to customize the shape even after the form is printed (Figure 2.4e). *ExpandFab* [50] prints mixed foam and elastic adhesive-based objects, and the novel technique expands the results up to 2.7 times in volume under heating (Figure 2.4f). Finally, Ko *et al.* [54] embedded thermoformable metamaterial cells into 3D models and modified the 3D shape post-printing. While these methods open new design opportunities for enhancing the behaviors of 3D-printed objects, additional ad-hoc setups (*e.g.*, environmental stimuli and actuation) or unique fabrication processes (*e.g.*, custom printing techniques) are needed to achieve desired behaviors. My dissertation aims to create structures that are 3D printable with readily available machines and materials, that can be easily integrated into 3D-printed objects during the printing process or assembled in the post-printing phase, and that can execute desired behaviors without specialized environmental configurations.

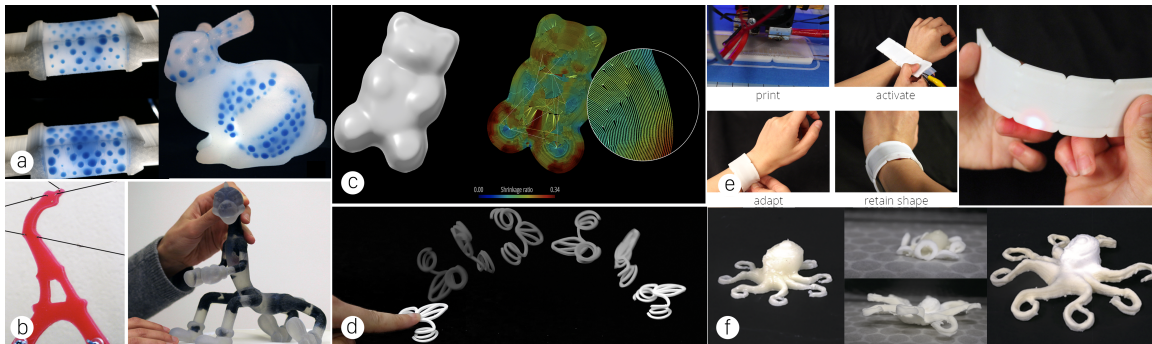


Figure 2.4: Deformable 3D printable objects made with elastic material and environmentally responsive materials.

2.3 Actuating 3D-Printed Objects for Interaction

3D printable kinetic objects can be further converted into interactive devices that perform desired movements [88, 108, 145]. To make actively movable 3D-printed objects, researchers have explored embedding external actuators such as electromotors [18, 61, 77, 63] and springs [56] or 3D printable actuating structures such as springs [23, 113] or pneumatic motors [28, 131]. Below, I describe these approaches and discuss their relevance of these approaches to this dissertation.

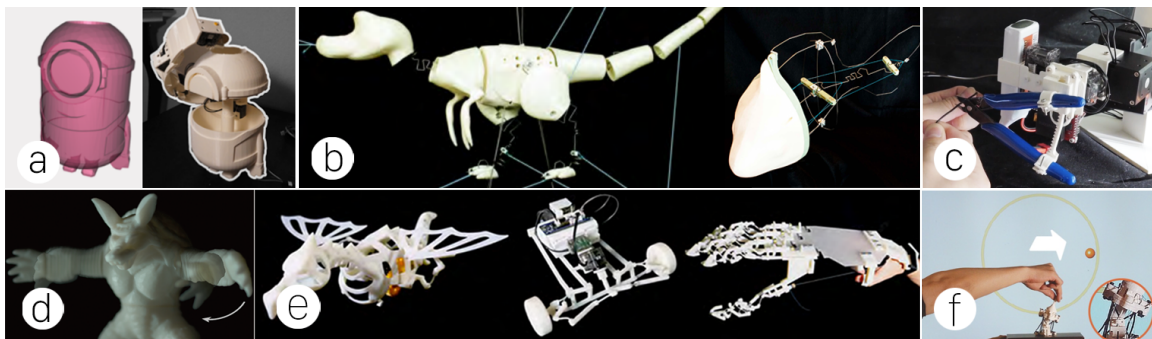


Figure 2.5: Making 3D printable kinetic objects for actuation using external motors and movable 3D-printed structures.

To actuate 3D-printed objects to move, prior work has coordinated external actuators such as motors [62, 72, 63] and pressure-controlled systems [67, 95, 133] with the 3D-printed

parts to control the object movements. Electromotors, one of the most used motor types, are connected to 3D-printed pieces to perform desired movements. For example, *Romeo* [61] converts 3D models into transformable robots for augmented functionalities with an embedded robotic arm controlled by external electromotors (Figure 2.5a). *Bend-it* [128] adds kinetic properties to 3D-printed models by connecting different 3D-printed parts with wires and controlling the movement of these parts with external motors (Figure 2.5b). *Roman* introduces a set of hardware designs that is 3D printable and attachable to everyday handheld objects, which are manipulated and actuated by conventional robotic arms (Figure 2.5c). Mechanical 3D-printed structures such as hair-like structures [82], hinges [73, 89], linkages [14, 48, 132], and kinematic pairs [2, 38, 75, 92, 109, 138] are also used to transmit the movement driven by external motors and even manual forces. For example, *Coded Skeleton* [45] heats the shape memory alloy embedded in a hinge-based 3D-printed body and deforms the structure in a controllable way (Figure 2.5d). Megaro *et al.* [72] introduced a computational pipeline to take as input an articulated compliant design and automatically replace joints with parametrizable linkage flexures. The optimal designs can perform versatile compliant movements when the linkages are driven and controlled by external motors (Figure 2.5e). *ReCompFig* uses 3D-printed compliant mechanisms and tensioning cables to create kinematic interfaces that dynamically provide multiple reconfigurable motional degrees of freedom (Figure 2.5f). Using an off-the-shelf spring motor, Song *et al.* [108] introduced a set of elemental mechanisms using cams and followers that can be 3D printable and embedded in wind-up toys of arbitrary shapes for expressive part motions such as translation and oscillation (Figure 2.6a). Li *et al.* [64] created customizable 3D-printed linkage mechanisms to provide movement and haptic simulations for proxies of virtual hand tools in VR (Figure 2.6b). Zhu *et al.* [145] created animated mechanical toys with computationally generated mechanical parts that perform linear and circular motion using cams and crank-sliders in a box beneath the toy characters (Figure 2.6c).

Besides external motors, researchers have also explored pressure-controlled systems as alternative actuators [131, 28] and embedding 3D printable actuators [121]. For example, MacCurdy *et al.* [68] introduced a novel technique that simultaneously deposits photopolymers and a non-curing liquid to fabricate complex and pre-filled fluidic channels and create

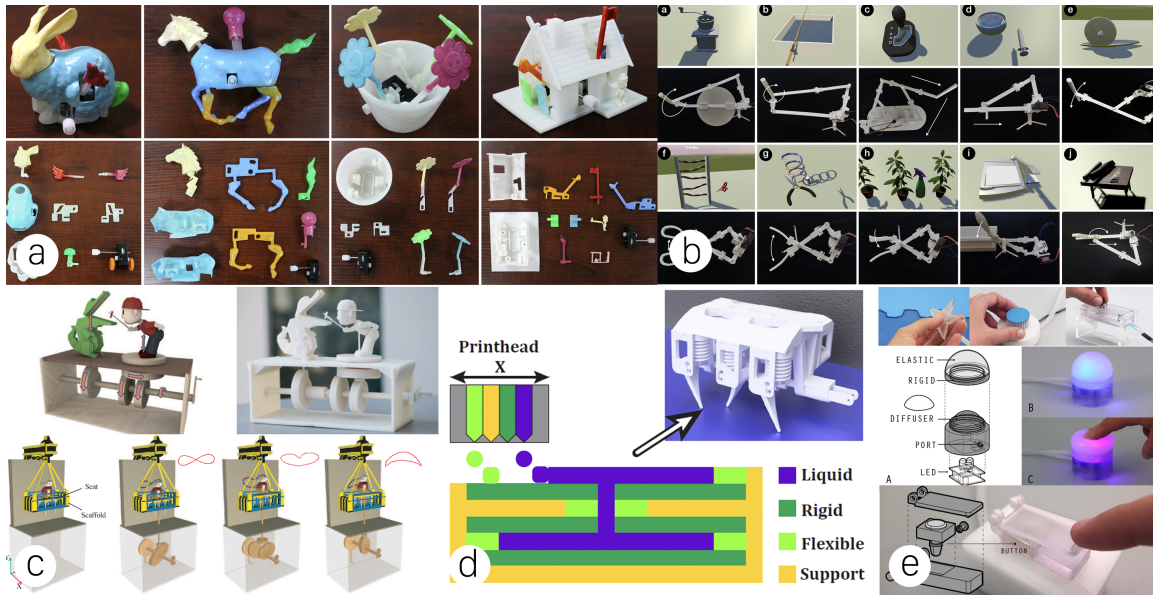


Figure 2.6: Making 3D printable kinetic objects for actuation using kinematic pairs and pressure-controlled actuator.

hydraulically actuated 3D-printed robots (Figure 2.6d). Vázquez *et al.* [122] created a series of 3D-printed pneumatic controls such as buttons and knobs that can detect user force actuation with air pressure sensors (Figure 2.6e). In addition, 3D-printed kinematic mechanical structures such as joints can be integrated into 3D objects for desired motion. For example, Ureta *et al.* [121] used revolute and ball joints to create complex mechanical objects that can execute simple joint-based motions such as hinge pivoting (Figure 2.7a). Besides joints, springs can store potential energy stored as the energy source for self-propelled movements [17, 113]. For example, 3D-printed bone-like springs are used as switches to release stored energy and produce driving force [23] (Figure 2.7b). Besides creating mechanical structures in the traditional FDM process, controlling novel fabrication processes can also produce 3D printable actuators. For example, Peng *et al.* [85] invented a 5-DOF FDM 3D printer that can wind copper wires in 3D-printed objects to make custom electromagnetic devices (Figure 2.7c). *A-line* [124] winds a straight line structure to form a jumping frog with helical spring-like legs when heated (Figure 2.4d). Lastly, *MorphingCircuit* [123] turns a flat substrate where functional electronics are assembled into interactive and programmed 3D

objects triggered by heating (Figure 2.7d).



Figure 2.7: 3D printable actuators are created for making 3D-printed objects move.

2.4 Adding Sensing Capabilities to 3D Printable Objects

3D printable kinetic objects can be further converted into interactive devices for sensing. To create 3D-printed objects for sensing, researchers have explored combining external sensors and electronics with deformable 3D-printed forms [20, 40, 90, 101, 123, 126, 130, 143, 22, 49, 76, 80, 97] and applying various sensing techniques to 3D-printed devices, such as acoustic sensing [58, 60], capacitive touch sensing [36, 111, 91, 139, 101, 39, 112, 99, 102], air pressure sensing [107, 118], motion sensing [127], and computer vision [96]. For the former method, Peng *et al.* [86], for example, created a custom machine that cuts soft fabric patches using a laser and then stacks layers of fabric to form a 3D shape. Electronics can be embedded into those soft and elastic 3D objects for interactions such as touch sensing (Figure 2.8a). *SurfCwit* [120] allows the user to design circuit routes on the surface of 3D-printed models and uses copper tape to construct conductive traces for connecting electronics. With built-in circuits, the 3D models can house sensors for interaction (Figure 2.8b). Similarly, *MorphSensor* [144] also allows for adding a circuit layer to the object’s surface using conductive ink jet printing and off-the-shelf electronics and sensor modules (Figure 2.8c). For the latter method, *Tactlets* [25], for example, creates tactile, touchable

controls on 3D-printed devices and adds sensing capabilities to those controls using capacitive sensing (Figure 2.8d). Recently, *MetaSense* [24] combines highly conductive printing material with 3D-printed metamaterial cell structures to sense the deformation based on capacitance changes (Figure 2.8e). Besides capacitive sensing, researchers also integrate other types of sensing techniques into 3D-printed objects for interaction. For example, *Lamello* [98] integrates comb-like structures into 3D-printed devices and uses a microphone to detect and interpret the passive acoustic signals generated by people striking the structures into interactive input (Figure 2.8f). *DefSense* [15] embeds piezoresistive sensing elements into flexible 3D printed objects and can continuously sense the deformation of the objects for run-time interactions (Figure 2.8g).

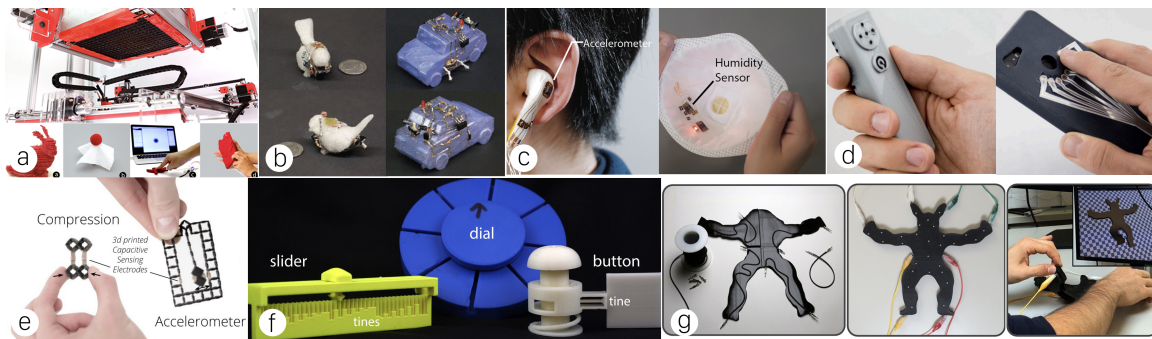


Figure 2.8: 3D printable objects use external sensors or integrate sensing techniques for interaction.

Deformable 3D printable structures can also be embedded to produce signals for sensing and data communication. For example, a helical spring (Figure 2.9a) can be embedded in a tablet stylus to render realistic tactile feedback, resulting in different writing pressures on touchscreens [114]. Ou *et al.* [82] invented a fabrication technique to 3D print hair-like structures on arbitrary geometric surfaces. Swiping on the hair structures produces different inaudible sound frequencies, which can be captured and identified as distinct user input (Figure 2.9b). *MechanoBeat* [117] is an electronics-free mechanical oscillator to monitor user interactions with daily objects using ultra-wideband array scanning. The vibration signals of the oscillator can be used to infer the user-object interactions (Figure 2.9c). Finally, Iyer *et al.* [47] introduce 3D-printed antennas that enable 3D-printed wireless sensors, input

widgets, and objects for communication via Wi-Fi (Figure 2.9d).

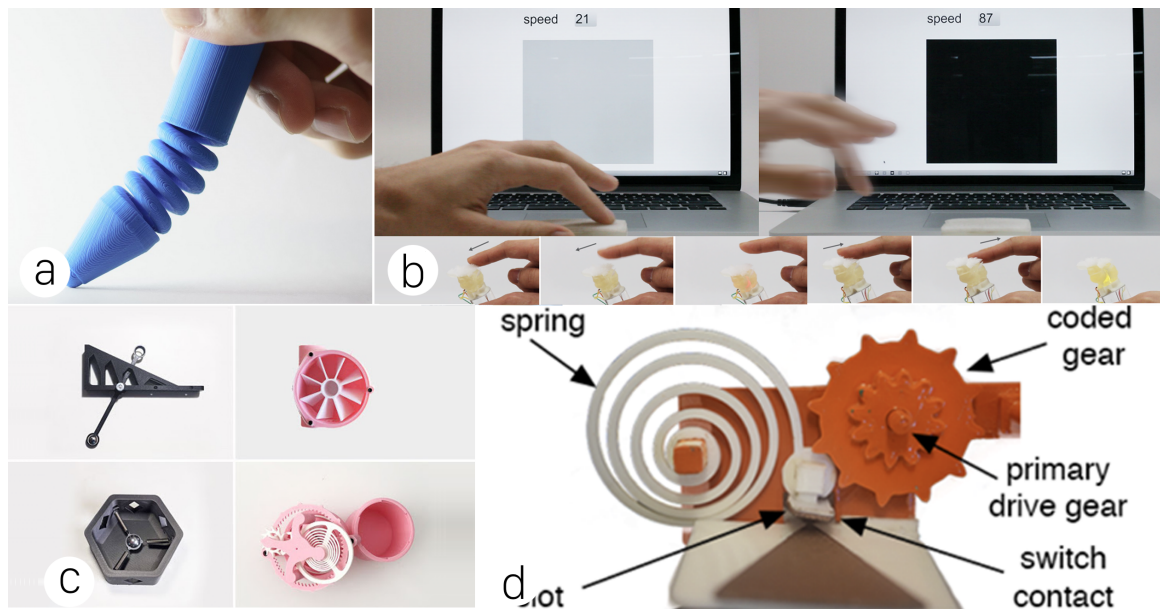


Figure 2.9: 3D printable objects use embedded kinetic structures for sensing.

In this dissertation, I investigated 3D printable springs as one of the primary mechanical components for both actuation and sensing. Unlike other kinematic components, springs store potential energy when deformed and are ideal as self-contained energy sources for triggering movements. For example, twisting a spiral spring to gain torsion and restoring the winded spiral spring to produce rotatory motion. Additionally, I also explored how to control the energy and the output movement by customizing spring parameters, which are rarely studied in prior work. With the expressive deformation behaviors of springs, I further studied how to empower 3D-printed objects with sensing capabilities through the structural deformation of various spring forms. Compared to other approaches of adding sensing capabilities to 3D-printed objects, my spring-based approach detects the capacitance changes caused by the spring deformation and interprets them as interactive user input.

2.5 Parametric Tools for Designing and Controlling 3D Printable Behaviors

As I create design techniques for different applications, another important part of this dissertation is to build interactive design tools, which allow the end-user to design and control desired kinetic 3D printable behaviors. One common type of parametric CAD tool allows users to use predefined modular structures from a template library to create functional 3D-printed objects from scratch [66, 92, 138]. The end-user can use provided designs to modify a 3D model for desired 3D printable deformation behaviors [16, 42, 44, 115]. For example, the design editor for adding *Metamaterial Mechanisms* cells to 3D models allows the end-user to select from a cell library (Figure 2.10a). With a preview of target behaviors in such CAD tools, the end-user can see the output motion of 3D models before they are fabricated [41, 121]. For example, the fundamental mechanisms used for providing expressive wind-up toy movements [108] are rendered to show the resulting movement in the design tool (Figure 2.10b). Similarly, the computational design system [138] also simulates the movements of all the kinematic parts repurposed into the custom 3D shape (Figure 2.10c). In [145], the designers can check the expected character motion is performed correctly with the selected kinematic elements through a simulation (Figure 2.10d). Finally, predefined structures can be used to construct 3D objects with extraordinary material and mechanical properties [57, 56, 104]. For example, Figure 2.10e shows that the user can edit and test the generated truss structures with provided primitives in the *TrussFormer* editor [55]. These truss structures can bear a certain amount of load or force.

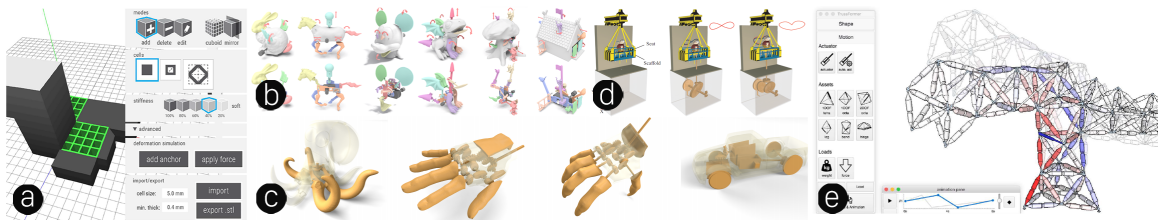


Figure 2.10: Applying predefined units from a template library for augmented behavior design in CAD tools.

Another common approach for 3D model customization is to directly edit an existing

model and iterate the embedded designs in-situ for target behaviors [94, 99, 103, 119, 135, 136]. Many intelligent design tools allow the end-user to input high-level specifications without knowing the underlying details and automatically generate the 3D design that meets the end user’s need [53, 71, 78, 103, 142]. For example, in *Romeo’s* design editor [61], the user can specify which part of the object can be transformed, how it moves in space, and the corresponding action to be taken (Figure 2.11a). In addition, the user can specify motion style and parameterize motion details for creating a 3D printable motion (Figure 2.11b) [52]. In the computational fabrication tool for creating compliant mechanisms [72], the user can parameter their desired oscillating motion and the links are generated based on the input motion (Figure 2.11c). When the user understands how to modify the 3D models for their desired behaviors, they select a region of the 3D model and convert it into a structure that accomplishes the design goals [99]. For example, in the tool for creating telescoping structure-based 3D models [134], the user can select a region along with the model skeleton and then customize the embedded telescoping structures in place (Figure 2.11d). Similarly, the user can specify what regions of the 3D object can have pop-up hinge mechanisms inserted (Figure 2.11e) [79]. Finally, when external components are added to existing 3D models, many design tools support the freedom of manual placement and orientation of those components [18, 25, 123, 143, 144] and synthesize enclosures for 3D printing [74]. For example, *PHUI-kit* [49] allows users to drag and drop physical widget models on curved 3D surfaces as with GUI components. The widgets can be repositioned with implementation details hidden beneath the surface (Figure 2.11f). *SurfCuit* [120] provides a side panel to show the circuit schematic while the user freely arranges the electrical parts in the 3D model (Figure 2.11g).

Inspired by past work, I summarize three key functions that are provided in the tools to embed spring-based mechanisms into 3D models for 3D printable kinetic behaviors: (i) properly situating spring-based mechanisms in custom 3D geometries, (ii) supporting direct edit and parameterization of the embedded spring-based mechanism, and (iii) displaying the updated 3D models in real-time for validation. In these tools, the user adjusts the high-level behavior characteristics, such as how the form changes and how much energy is contained, and the user input is converted to the underlying geometric design of the embedded spring-

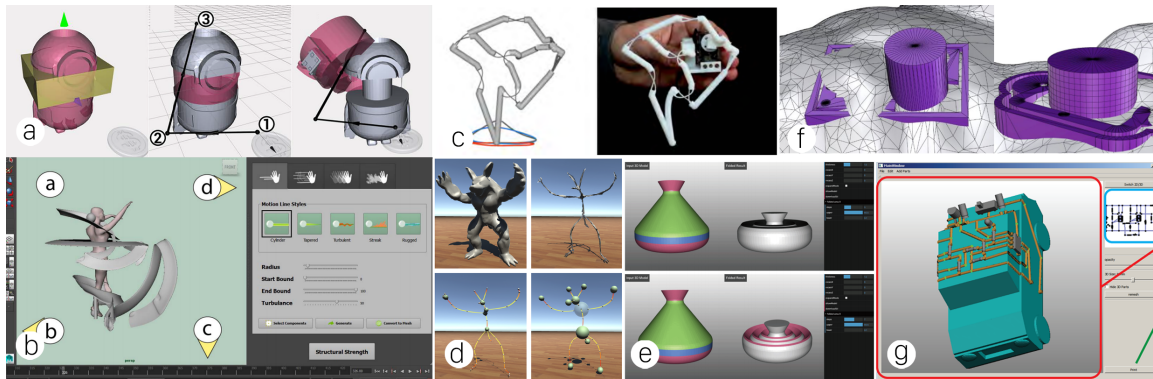


Figure 2.11: Direct editing 3D models for desired behaviors in CAD tools.

based mechanisms automatically. To accomplish all the computation and generation, spring theory and principles are built in these interactive design tools.

2.6 Chapter Summary

In this chapter, I first provided background on the physics and theory of mechanical springs from the field of physics and mechanical engineering. Then, I recognized the benefits offered by spring structures and highlighted three spring types that are parameterizable and have the potential for achieving various kinetic behaviors for interaction in 3D printing.

I then reviewed the literature on making 3D printable kinetic objects for desired deformation behaviors, controllable movements, and sensing user input for interactions. For creating deformable 3D-printed objects, I identified two common approaches—embedding 3D printable kinematic mechanical elements in 3D models and making shape-changing objects with environmental reactive material composites. My dissertation explores springs, which are not widely studied for 3D printing, for making kinetic objects. I also noticed that springs are ideal energy sources to triggering movements, while using external motors and electronics or manually actuate objects to move are mostly seen in prior work. Lastly, informed by previous work on adding sensing capabilities to 3D-printed objects, I further identified new technique that combines capacitive sensing with various embedded spring forms to achieve user input sensing through spring’s structural deformation, which provokes

novel sensing applications for 3D printable spring-based objects.

Finally, I also related my work in this dissertation to the literature on the development of custom design tools, which were created to lower barriers for the end-user to design and control parametric 3D models. By categorizing design tools for specific purposes, I drew commonalities and insights from prior work to inform the design and development of such interactive tools, with a focus on how to enable 3D modelers to integrate, customize, and validate novel spring-based mechanisms in 3D models for different kinetic behaviors. These key characteristics are reflected in the series of custom design tools that I built in my research—from Chapter 4 to Chapter 6.

Chapter 3

MAKING THINGS MOVE: UNDERSTANDING HOW 3D PRINTABLE KINETIC OBJECTS ARE DESIGNED AND CREATED

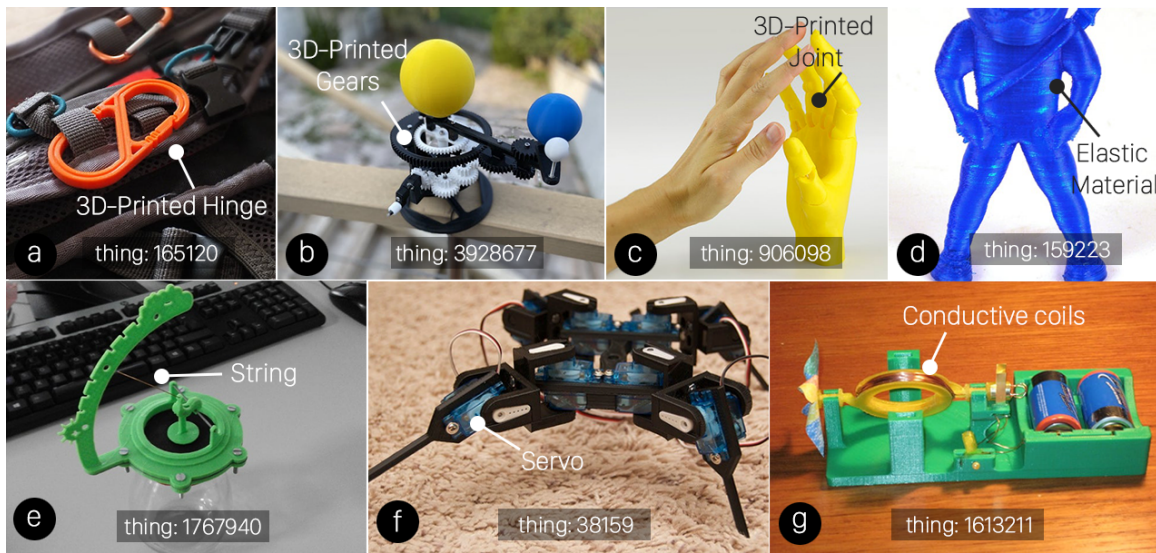


Figure 3.1: A set of example 3D-printed kinetic designs found on Thingiverse showcase the usage of integrated 3D-printable kinematic mechanisms such as (a) hinges, (b) gears, and (c) joints, or (d) elastic material, or external hardware such as (e) strings, (f) servos, and (g) electronics for kinetic behaviors.

In the past decades, 3D printers have become cheaper and more accessible to transform rapid prototyping and enable personalized fabrication [7, 6, 69]. And yet, most 3D-printed objects created with consumer-grade 3D printers are static and rigid [8, 141]. Why might this be? Is this a representation of maker’s true creative interests or a limitation of our 3D-printing tools?

In our work, we argue that it is the latter. First, we have seen skilled 3D printing enthusiasts create complex kinetic models such as clocks, robots, and prosthetic devices and share them in online CAD repositories like Thingiverse (Figure 3.1). Second, fabrication

researchers have identified limitations in current CAD tools and 3D printing workflows and have tried to create 3D printable kinetic designs that bend [45, 89, 94], rotate [72, 138, 145], move [41, 68, 108], or otherwise deform [16, 83, 124] in controllable ways. Motivated by these two emerging trends in the maker community and the fabrication research, I questioned: *what kinetic 3D printable designs are makers currently making? How are they creating those kinetic designs? What are the challenges and opportunities for future makers and fabrication researchers?*

To answer these questions, I looked at the uploaded creations on *Thingiverse*—the largest and most popular online 3D printing and CAD design repository—and analyzed the collected 3D printable kinetic designs to identify merging standard practices and themes for making kinetic 3D-printed objects. In this chapter, I describe this large-scale study—Making Things Move (MTM) [30]. Using keyword search methods similar to [5, 12], we collected and qualitatively analyzed 1337 kinetic creations uploaded to Thingiverse. We explore the making purpose/category, characteristics of the movable parts, the resulting kinetic behaviors, actuation methods, and the design and making processes. Our results show that most kinetic designs were toys and sculptures. Joints, hinges, and gears were the most common for creating kinetic designs. Rotations were dominant amongst other output behaviors performed by created kinetic objects. Finally, electro-motors and flexible components such as springs and rubber bands were also common as actuators besides manual actuation. Our data allows us to deeply reflect on the making practices of 3D printable kinetic objects. Based on the reflection, we also identify common creation patterns and the gap between what makers create and what fabrication research achieves, including challenges and opportunities for makers to create custom 3D-printed kinetic objects.

This chapter contributes: (i) characterization of current trends in making 3D printable kinetic objects, including the purposes for making, the used mechanisms for movements, actuation methods, design and making processes, and kinetic behavior types that are primarily seen in the creations; (ii) investigation of the creation patterns for making kinetic designs on Thingiverse and the gap between practical making and fabrication research on creating kinetic 3D-printed devices; and (iii) an open-source dataset of representative kinetic designs on Thingiverse with our applied qualitative codes and metadata, enabling future

Table 3.1: Search terms related to kinetic mechanisms and kinetic behaviors and their frequencies in the final complete set.

Kinetic Mechanism-Related Term (Frequency)	Kinetic Behavior-Related Term (Frequency)
Gear (128), crank (80), hinge (75), spring (69), joint (51), flexi (46), linkage (46), fabric (35), bearing (23), telescoping (7), lattice (7)	Moving (194), mechanical (103), flexible (102), articulated (84), movable (68), print in place (48), movement (45), kinetic (41), bendable (31), snap (15), compliant (13), action (10), movable design (9), behavior (7)

meta-analyses by the fabrication community.

3.1 Method

To examine 3D printable kinetic designs and the making practice on Thingiverse, we manually searched for models using an iterative keyword list, then qualitatively reviewed and coded search results to create a study dataset, and finally analyzed patterns such as kinetic mechanisms, behavior types, and actuation methods.

3.1.1 Data Generation

Similar to [5, 12], we first generated a list of search terms related to kinetic mechanisms and behaviors, which were derived from the fabrication literature and colloquial words (Table 3.1). Then, we searched for relevant 3D designs on Thingiverse using individual and combined terms from both lists (multi-word terms were closed in quotation mark pairs). Since all Thingiverse results were sorted by relevance, we stopped examining items when relevant results no longer existed or became sparse (*e.g.*, fewer than three relevant designs shown on five successive result pages).

After initial queries, we refined our term list by adding newly found relevant keywords and removing terms that yielded irrelevant or repeated results. First, we extracted frequent terms that described kinetic models but were not included in our initial list from the textual descriptions on the model page and the name of the collection (if available) to which the model belonged. For example, “Flexi” is typical for a “flexible” 3D design collection. We

also added terms that described specialized printing approaches and implied kinetic designs such as “print in place”. Then, we removed terms that yielded irrelevant results. For example, most results yielded by “kinematic” were rigid parts for kinematic devices such as robots and 3D printer accessories. For the plural and singular forms of a kinematic mechanism (*e.g.*, “joints” and “joint”), we merged the results yielded by both terms but used the singular form to represent this mechanism in our final dataset. In addition, we kept one of those terms that described the same thing. For example, the search results for “compliant mechanism” and “compliant” were highly overlapping. Lastly, if a term A described the behavior performed by a term B or term B represented a sub-category of term A, we kept the term A and excluded term B. For example, a “buckle” performs “snap” and “bistable” designs representing a subset of “compliant” devices.

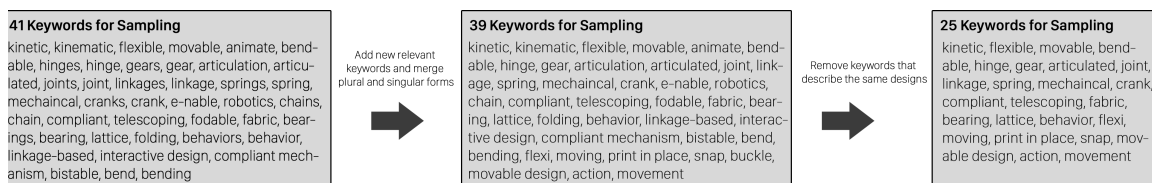


Figure 3.2: The iterative process for refining the search term list.

Finally, we generate a complete set of 2139 3D printable results with the refined search terms^{3.2}. However, the generated dataset contained redundant kinetic designs due to overlapping results generated by different search queries. Consequently, we cleaned our dataset using the following exclusion criteria: (i) if two results are the same design (*i.e.*, sharing the same URL), exclude one; (ii) if two designs have a direct remixing relationship through the Thingiverse remixing mechanism and the remixer does not add new designs for a different purpose, exclude the remixer design (Figure 3.3a); and (iii) If two designs have no direct remixing relationship and they share the exact mechanism for the same purpose, exclude one regardless of their different appearances (Figure 3.3b).



Figure 3.3: Examples of identical 3D designs in our dataset: (a) a set of remixed bearing designs without additional new designs and (b) different flexible animal figurines with the hinge-based design. The check marks mean the creations are selected to store in our dataset, while the X marks indicate those are not included.

3.1.2 Qualitative Analysis of 3D Kinetic Designs

We qualitatively analyzed our Thingiverse dataset using an iterative coding process (similar to [5]). Appendix A shows the final codebook with 18 codes across five high-level categories: (1) *Model Category/Purpose*; (2) *Kinetic Component Design*; (2) *Kinetic Behavior*; (3) *Design and Fabrication Process*; (4) and *Social Interaction*. For analysis, we examined Thingiverse model pages for each design, which includes open-ended description fields, images/videos, closed-form data on the printing process (*e.g.*, 3D printer, resolution, infill density), social interactions (*e.g.*, likes, remixes), and other metadata (*e.g.*, Thingiverse category).

To assess the model itself and the underlying kinetic mechanisms and behaviors, we reviewed uploaded images and/or videos (if available), the uploader’s textual descriptions, and by examining the CAD model in *Customizer*—a built-in web app on Thingiverse. In a small number of cases, we also downloaded CAD models and analyzed them in the slicing software—*Cura*. For the *Kinetic Component Design* and *Kinetic Behavior* codes, we began with an initial list drawn from literature [2, 6], which was iteratively expanded as necessary.

For the iterative coding process, two researchers began by independently coding two sets of 30 randomly selected designs using an initial codebook (with 12 dimensions) and then met to discuss and resolve disagreements through consensus and updated the codebook as necessary. This process was repeated for five rounds (241 designs, 18% of our dataset) until the codebook solidified and inter-rater reliability was reached—a *Cohen’s Kappa* average of 0.83 ($SD=0.14$) across all the 18 dimensions. Finally, one researcher completed coding the

remaining dataset using the finalized codebook.

3.2 Findings

3.2.1 Overall Trends

Our final dataset contains 1337 3D printable kinetic designs uploaded by 1056 unique creators. Across the 11 mutually-exclusive application categories defined by Thingiverse, the most common designs included *Toys & Games* (26.1%, N=349), followed by *Hobby* (10.2%, 136), *Learning* (9.2%, 123), and *Models* (9%, 120)—see Table 3.2 and Figure 3.5. To achieve kinetic behaviors, most designs (66%, 882) used purely 3D printable mechanisms such as joints or hinges, and 29.3% (392) combined both 3D printable mechanisms and external non-printable hardware such as string, band, rods, steel springs, or electronics, and 3% (40) relied solely on elastic printing materials such as NinjaFlex or TPU, and 1.7% (23) exclusively used external hardware. Figure 3.4 shows a decomposition of the dataset based on used components and materials. For the kinematics themselves, motion behaviors were most common such as rotating and translating (79.8%, 1067), then deformations like bending and folding (60.5%, 809). In addition, we found that 949 models (71%) required some form of post-processing, such as manual assembly, sanding, or gluing.

Table 3.2: Frequencies of 3D-printable kinetic designs across 11 Thingiverse-defined categories.

Category	Count	%	Description
Toys & Games	349	26.1	Mechanical toys, toy & game accessories, chess, construction toys, dice, games, playsets, puzzles.
Hobby	136	10.2	DIY, robotics, sport & outdoors, R/C vehicles, music, electronics, automotive.
Learning	123	9.2	Engineering, math, biology, physics & astronomy.
Models	108	8.1	Animals, buildings & structures, creatures, food & drink, model furniture, model robots, people, props, vehicles.
Fashion	96	7.2	Accessories, costume, bracelets, earrings, glasses, jewelry, keychains, rings.
3D Printing	96	7.2	3D printer accessories, 3D printer extruders, 3D printer parts, 3D printers, 3D printing tests.
Art	93	7.0	Sculptures, interactive art, art tools, 2D art, coins & badges, math art, scans & replicas, signs & logos.
Tools	93	7.0	Machine tools, hand tools, tool holders & boxes.
Household	93	7.0	Décor, bathroom, containers, household supplies, kitchen & dining, office organization, outdoor & garden, pets.
Gadgets	52	3.9	Computer, mobile phone, tablet, video games, camera, audio.
Other	98	7.3	Misc not covered by other categories.
Total	1337	100	

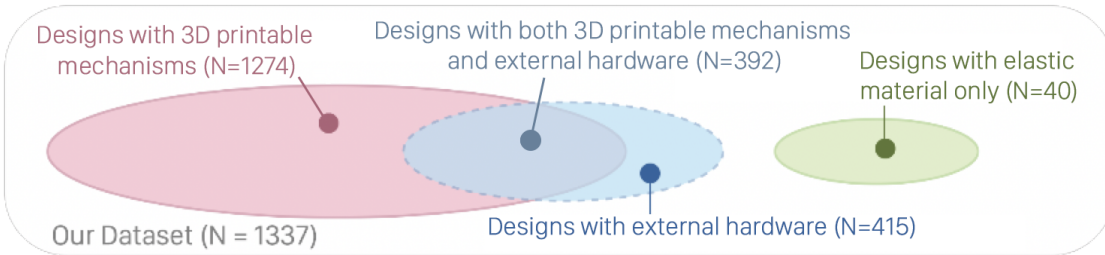


Figure 3.4: The final dataset consists of designs that use 3D-printable mechanisms, external hardware, and purely elastic materials.

Thingiverse operates not just as a repository but as a social platform; we also studied *likes*, *collections*, *makes*, and *remixes* amongst users. Of the 1337 models, over 76.1% (1017) received some social interaction, 63.9% (854) were marked as downloaded and printed by another user, and 36.1% (482) were remixed. Relatedly, 1079 models (80.7%) were started “from scratch” while 258 (19.3%) were marked as remixing other Thingiverse models.

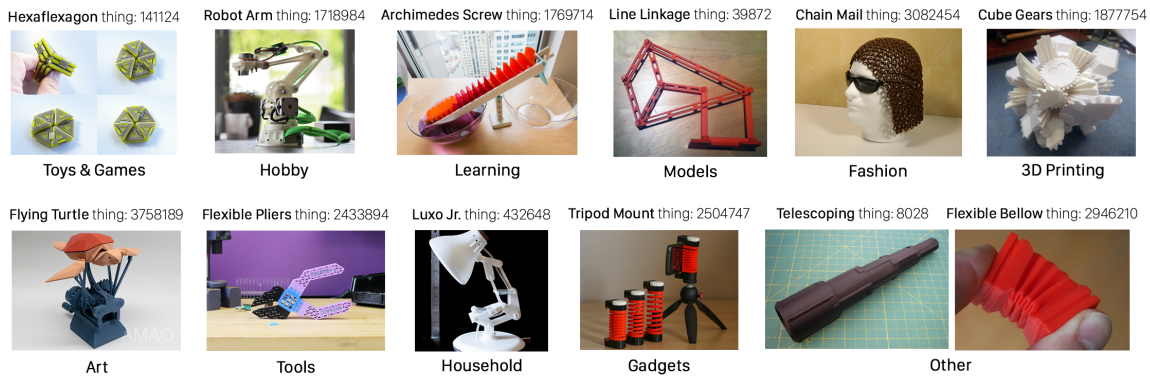


Figure 3.5: Examples of 3D kinetic designs across 11 application categories defined by Thingiverse.

3.2.2 3D Printable Kinetic Mechanisms

To understand how makers create 3D printable kinetic objects, we examined *which* 3D printable kinetic mechanisms and *how often* they are used. Of the 1337 designs, nearly all

(97%, N=1297) included either 3D printable kinetic mechanisms designed for rigid plastic such as PLA or PETG or external kinematic hardware, while some models (3%, 40) relied solely on elastic material to achieve movements (*e.g.*, Ninjaflex or TPU). Of those 1297 designs, 98.2% (1274) used 3D printable kinetic mechanisms, and 32% (415) used external components (Figure 3.4). Of the 1274 designs, joints (36.5%, 465) were most common—perhaps because joints are relatively easy to model and different types of joints (*e.g.*, universal joints and ball joints) provide expressive movements—followed by hinges (29.5%, 376), gears (26.6%, 339), linkages (14.8%, 189), and cranks (12.9%, 164). See Figure 3.6 for some examples of 3D printable kinetic mechanisms used in 3D designs. The full frequency distribution of 3D printable kinetic mechanisms is: joint (36.5%, 465), hinge (29.5%, 376), gear (26.6%, 339), linkages (14.8%, 189), crank (12.9%, 164), spring (8.8%, 112), axel (8.1%, 103), interlocking (6.8%, 87), lever (6.2%, 79), bearing (5.9%, 75), microstructure (5.1%, 65), cam (2.4%, 30), telescoping (2.3%, 29), slider (1.9%, 24), and other (1%, 13).



Figure 3.6: Examples of 3D printable kinetic mechanisms.

We also examined co-occurrence patterns across kinetic mechanisms: how often are mechanisms combined and with what other structures? We found that single-mechanism designs¹(54.1%, 689) were more common than multi-mechanism designs (45.9%, 585). Of

¹The same design can be used once or multiple times in a single-mechanism design.

those 689 single-mechanism designs, hinges were most prevalent (33.7%, 232), followed by joints (25.8%, 178), gears (10.4%, 72), and springs (6.1%, 42). For multi-mechanism designs, most commonly, makers combined joints and gears (25.3%, 148), followed by gears and cranks (16.6%, 97), joints and linkages (14.7%, 86), and hinges and linkages (9.6%, 56). These combinations of kinematic components not only show the frequency of elements used for making kinetic objects but also inform what and how kinematic elements are often paired to support kinetic behaviors in 3D-printed designs.

Finally, we also examined how different kinematic mechanisms support kinetic applications in varying domains. When examining the usage of mechanisms as a function of the Thingiverse category, we found that joints were the dominant mechanism used in 4 categories (*i.e.*, *Toys & Game*, *Models*, *3D Printing*, and *Tools*), while hinges surpassed joints in 3 categories (*i.e.*, *Hobby*, *Fashion*, and *Household*) and gears became the most used mechanism in 3 categories (*Learning*, *Art*, and *Gadgets*). Across all the categories, most of the kinetic mechanisms were used for *Toys & Games*; however, *Learning* was the dominant application category for microstructure, and most interlocking structures were used for creating *Fashion*. For example, a creator learned about the 2-DOF compliant space pointing mechanism by reading a related academic paper and experimented with the compliant design as a learning process (thing: 3612786). These findings allow us to select proper kinetic mechanism candidates for specific application categories.

For those kinetic designs that use springs, beam-based springs are most prevalent (50.9%, 57), followed by spiral springs and helical springs. Most beam-based springs are embedded into toys and gadgets to provide shape-changing abilities, while helical springs are created as decorations, such as a spring ball (thing: 3162270) and a Christmas tree (thing: 3135231), and spiral springs are found in many applications where continuous turning occurs, such as a clock (thing: 3364860) and a pull-back car (thing: 3308710).

3.2.3 3D Printable Kinetic Behaviors

In addition to kinetic mechanisms, we also examined kinetic behaviors such as rotations and bending. While related, similar output behaviors can be achieved with different mechanical

designs and/or materials; thus, they are worth examining independently. For example, a rotary catapult arm can be made of a spring (Figure 3.7-thing:3662245) or a lever (Figure 3.7-thing:1860072) and interlocking chainmail (Figure 3.6-thing:3096598), which simulates the elasticity of fabric. Of the 1274 models that used 3D printable kinetic mechanisms, rotations were most common (79.4%, $N=1012$), followed by bending (27%, 344), articulating (24.9%, 317), translating (18.7%, 238), and twisting (17.8%, 227). The full frequency distribution of 3D printable kinetic behaviors is: rotate (9.4%, 1012), bend (27%, 344), articulate (24.9%, 317), translate (18.7%, 238), twist (17.8%, 227), compress (13.8%, 176), fold (8.4%,108), stretch (7.1%, 91), oscillate/reciprocate (4.9%, 62), transform (4.6%, 59). Figure 3.7 shows examples of 3D printable kinetic behaviors.



Figure 3.7: Examples of 3D printable kinetic behaviors.

Similar to our mechanism analysis, we also studied co-occurrence behavioral patterns to understand what and how combinations of behaviors can be embedded into 3D kinetic designs. Most designs could perform multiple output behaviors (62.6%, 798), including articulating and rotations (36.5%, 291), bending and rotations (25.9%, 207), and translation and rotations (23.4%, 187). For example, an articulated robot could rotate its arms (Figure 3.7-thing: 35752), a rotating crank bent a lever to make some noise in a noise gun (Figure 3.7-thing:1167907), and a spring-loaded boat moved forward when the spring and gears rotate (Figure 3.7-thing: 3500845). Of the 476 single-behavior designs, most performed

rotations (77.9%, 371), bending (7.8%, 37), and compression (4.8%, 27). For example, a mechanical iris opened under manual rotation (Figure 3.7-thing:997182), a hinge-based crossbow bent when the string was loaded (Figure 3.7-thing: 2020668), and the user opened a clip by compressing a hinge-based handler (Figure 3.7-thing:2988949). Informed from the designs that perform multiple behaviors or only one single behavior, rotation surpasses other motion types and becomes the most popular movement people are making kinetic objects for. It is worthwhile to further explore how these rotations are enabled by kinematic mechanisms and how they are designed in accompany with other output motion types.

We also examined the relationship between kinetic behaviors and mechanisms. Of the 688 single-mechanism designs, 36.5% (251) produced a single output behavior—*e.g.*, a hinge-base carabiner resulted in only bending (Figure 3.1a) and an axel-based pin display resulted in only pin translations (Figure 3.8-thing: 3402410)—while 63.5% (437) achieved multiple distinct behaviors—*e.g.*, a ball joint-based solder helper hand performed both articulation and rotation (Figure 3.6-thing: 2487181) and a linkage-based compliant mechanism could produce oscillation, rotation, and transformation (Figure 3.5-thing:39872).

For those 586 multi-mechanism designs, 61.6% (361) resulted in multiple kinetic behaviors—such as torsion springs and gears in a 3D printable wind-up car (Figure 3.8-thing: 3308710)—while 38.4% (225) used multiple mechanisms to achieve a single behavior, *e.g.*, a hand cranking flashlight only included rotation while gears and cranks were used (Figure 3.6-thing: 13820).

In particular, bending is the primary output behavior for those designs that use springs as the key mechanisms (37.5%, 42) because beam springs and a few helical springs are used for bending movement. Helical springs are also used for creating compressions, such as the clamp (thing: 2988949) and the battery holder (thing: 374802). Spiral springs are always used for winding and therefore, rotary movement.

3.2.4 Use of Elastic Materials

Besides 3D printable kinetic mechanisms, makers also used external hardware (*e.g.*, elastic string, band, rods, steel springs, or electronics) or purely elastic printing materials (*e.g.*,

TPU) to achieve desired output behaviors. Of 415 models that included external hardware, most (94.5%, 392) still used some form of 3D printable kinetic mechanisms, and others (5.5%, 23) used the external actuator (*e.g.*, electromotors) or kinetic material (*e.g.*, fishnet, string, elastic steel wire).

For those 40 designs created with pure elastic printing materials, most (85%, 34) performed bending, such as bracelets and watchbands, followed by stretching (25%, 10), such as shoelaces or stretchable toys. Since no specific mechanisms were embedded, the created objects performed freeform deformations. For example, a Ninja doll made with flexible NinjaFlex material could be stretched, twisted, and bent freely. Additionally, we found that some kinetic mechanisms were printed using the elastic mechanisms to offer more flexibility for deformation or produce more frictions. For example, a tank chain used a chain of hinges printed with elastic materials and produced more friction on the ground.

3.2.5 Actuation Types

To activate kinetic mechanisms for desired behaviors, we examined how makers operate and control 3D-printed objects. Of the 1337 designs, most (85.3%, 1141) were activated by human power such as hand cranking or pulling, and 12.3% (164) were triggered by non-manual power sources such as electromotor or spring, and 2.4% (32) used both approaches. For those 32 designs, manual operations and non-manual actuators were usually interchangeable. For example, a set of gyroscopic cube gears could be actuated by hand cranking or an external electromotor (Figure 3.8-thing: 338808). However, 65.6% of the 32 designs also used external energy-charging components such as springs and rubber bands. These components charged energy through human operations and then released the stored energy for an output motion. For example, a 3D-printed boat is powered by a loaded spiral spring winded manually for energy charging.

We further examined the common actuation method in both manual (87.7%, 1173) and non-manual (14.7%, 196) actuators. Of 1173 manual actuation, rotating (36.3%, 426) was most common—perhaps because the most common kinetic mechanisms (*i.e.*, joints and gears) in the dataset required rotary operations (36.3%, 426) to trigger movement—followed

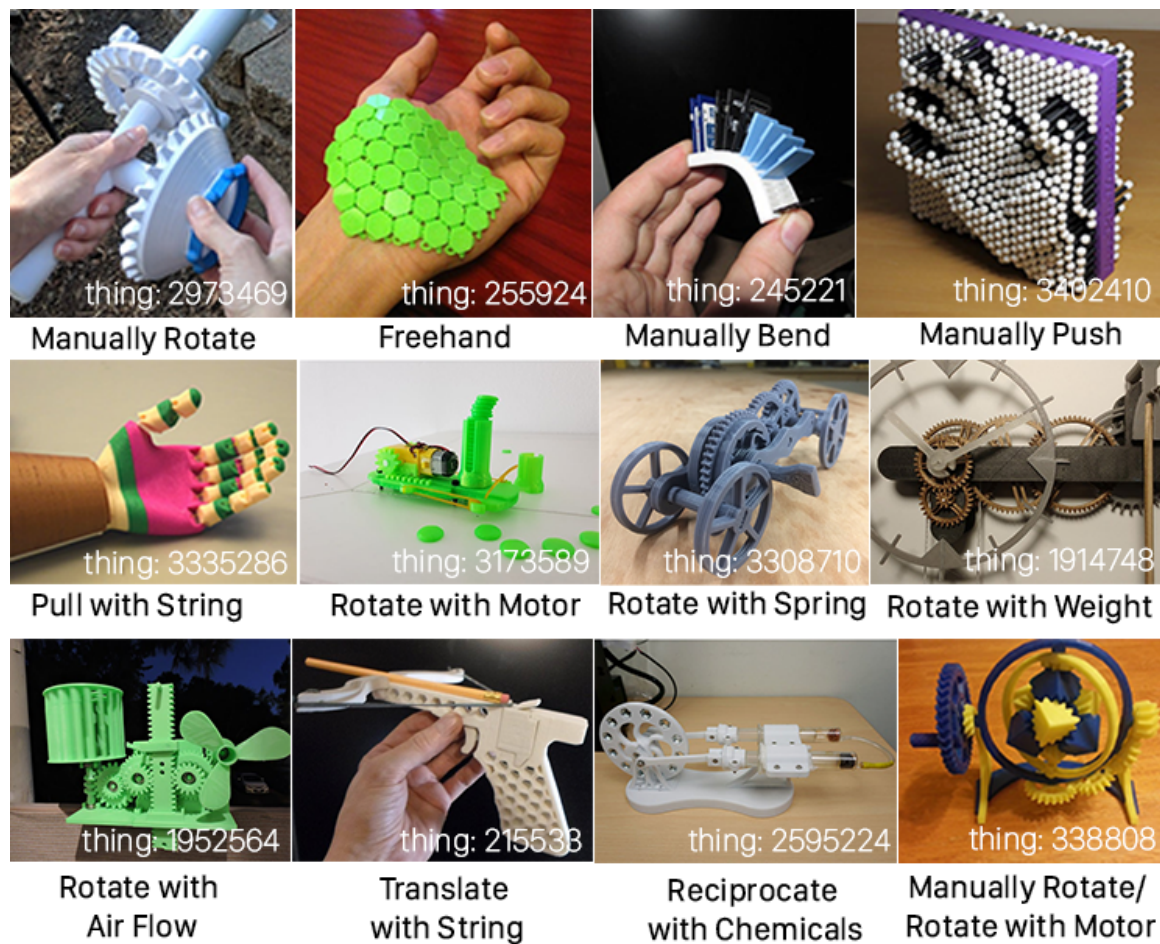


Figure 3.8: Examples of actuation methods.

by freehand operating (24.6%, 288) and bending (19%, 223). For example, the user rotated the roaster for fishing (Figure 3.8-thing:2973469), a chainmail-based fabric sheet deformed under freehand operations (Figure 3.8-thing: 255924), and bent a flexible SD card holder to pick the card (Figure 3.8-thing: 245221). Although we categorized manual actuation in different actions, the control of the output behaviors was unclear and not explicitly described by the creators. However, detailed instructions were provided for those applications that required specific operations. For example, to operate a prosthetic properly (Figure 3.8-thing: 3335286), the user needed to wear the device carefully and follow instructions to pull the string that triggered the finger movement. Compared with manual actuation, non-manual

actuators were used in a controllable way to start the action. Of 196 non-manually triggered designs, more than half (57.7%, 113) used electro-motors such as DC motors and servos to program the output movement. In contrast, other actuators such as springs (13.3%, 26), weight (7.7%, 15), and airflow (7.7%, 15) were used in specialized ways for behavior control. For example, a spring-loaded pull-back car (Figure 3.8-thing: 3308710), a 3D-printed clock used the oscillation of weight to function (Figure 3.8-thing: 1914748), and a fan turned under wind blowing (Figure 3.8-thing: 1952564).

3.2.6 Design and Making Processes

We also studied how 3D-printed kinetic objects were designed, created, and shared on Thingiverse.

Documentation of Kinetic Design on Thingiverse. In our dataset, we found that 92.1% (1231) designs contained some form of documentation—including design tools, printing-related information, instructions for post-print processes, and iterations, while 7.9% (106) were undocumented on Thingiverse. Of the 1231 documented models, most (88.5%, 1089) included printing settings (*e.g.*, resolution, infill density, printer brands, or support need). In addition, nearly half (49.6%, 610) described the instructions for post-processing (*e.g.*, assembly instructions and crafting methods for polishing the prints), 42.2% (519) covered the used design tool, and 35% (431) mentioned model iterations in the *Summary* filed on the model page. Furthermore, by qualitatively analyzing each design’s description (if available), we found that Thingiverse users documented the processes for making the kinetic design for several reasons: (1) for public sharing. For example, “*Here, I’m sharing what I’ve learned about clock gears*” (quote from the summary of the thing: 1282677); (2) for recording updates. For example, “*check back for updates as I revise the files*” (quote from the summary of the thing: 9863); (3) for self-advertisement. For example, “*... feel free to take a look at my other designs...*” (quote from the summary of the thing: 3723618); and finally, (4) for educational purposes. For example, “*... I did this as an educational project for kids or people new to 3D printing...*” (quote from the summary of the thing: 3164774).

Design Techniques and Tools. We found a similar trend of remixing and re-making in both designs from scratch and remixed from others: makers are more likely to re-creating existing kinetic objects than make new kinetic designs. For those created from scratch, we found that 66.3% (715) of original models were remixed or re-made by other makers, while 33.7% (364) were not. The 715 designs presented a breakdown of 48.5% (347) only re-made by others, 46.2% (330) both remixed and re-made, and 5.3% (38) only remixed. For those 258 unoriginal designs, more (39.1%, 101) were both remixed and re-made by others than either re-made only (29.5%, 76) and remixed only (5%, 13), while 26.4% (68) had no remixing and re-making activities. Amongst all the kinetic designs, the most popular original kinetic design re-made and remixed by other makers was a customizable, flexible bracelet (thing: 46735; the sum of makes and remixes was 1172), while a gear bearing (thing: 53451; No. of makes was 843 and No. of remixes was 6228) was the most common for both makes and remixes. Finally, only 38.7% (517) described the design tools for 3D modeling, including *Autodesk Fusion 360* (19.7%, 102), *OpenSCAD* (17.2%, 89), *TinkerCAD* (10.6%, 55), *Solidworks* (9.3%, 48), and a variety of commercial CAD software (43.1%, 223) such as *Blender* (6%, 31), *Sketchup* (5.8%, 30), *Autodesk 123D* (4.3%, 22), *Rhino3D* (3.9%, 20), and *Inventor* (3.9%, 20). Most commonly, creators complained about the lack of support to enable them to create the desired mechanism. For example, “... *I tried for a long time to figure out how to create a better-looking box when I stumbled upon a linkage design tool.*” (quote from the summary of the thing: 3006492). To create springs, there is no information about the design process and tool support but some makers posted suggestions for printing, such as printing a helical spring (thing: 3870039). Most 3D printable spring examples are made for demonstrations of 3D printing springs, either created from scratch or existing 3D models.

Post-Print Processes. Of the 1337 designs, 71% (949) required some forms of post-print processes such as assembly, gluing, sanding, and lubricating. Amongst various post-print methods, assembly is applied to nearly all designs (97%, 921), followed by gluing (15.6%, 148), circuit building (9.5%, 90), sanding or lubricating (10.5%, 100), and cutting (6.8%, 65). To achieve a specific function, makers also used other creative methods as part of the post-processing. For example, to convert a segment of a printing filament into

a hinge latch, the maker melted down one side of some filament to form a flat head like a nail (thing: 627077). Besides those (43.7%, 415) used external hardware, 532 (56.1%) designs that purely used 3D printable kinetic mechanisms still required post-processing. For example, the maker used flexible clips to snap rigid legs for a printable strandbeest (thing: 3073375), and another maker applied silicone-based oil to lubricate the cylinder walls, piston wrist, and crankshaft bearing surfaces in an engine model (thing: 1195361). For those with embedded springs, the springs are usually printed with support materials and carefully cleaned by additional manual work post-printing.

3.3 Discussion: Maker Community V.S. Fabrication Research

Our study showed that makers create 3D printable kinetic objects and share their designs within the community. While these findings reflect the common desire for making kinetic 3D-printed applications within the maker community, there is still room for improvement in the design space of 3D printable kinetic applications, including the complexity of kinetic designs, design aids for making such kinetic devices, and novel techniques introduced by fabrication research. In this section, I discuss the commonalities and differences in making 3D printable kinetic designs between the maker community and fabrication research, as well as the opportunities for using springs in both communities.

Makers and fabrication researchers used similar approaches to augment 3D-printed with kinetic behaviors. First, 3D printable kinetic mechanisms were studied and used in both making practices and research. For example, an articulated hand was created using joints for expressive finger movements (Figure 3.1c), and an assembly-free articulated forearm was demonstrated by leveraging the friction in ball joints [16]. Telescoping structures were used for creating an extendable backscratcher (Figure 3.6-thing: 3125608) and a resizable character [134]. In *Digital Mechanical Metamaterials* [44], bistable springs were used to keep and restore states to control digital information. Bistable springs are also used to present different shapes in compliant mechanisms in the maker community, such as thing: 3163115. Combining both 3D-printed parts and external hardware was also found in both communities. For example, fabric and 3D-printed rigid parts were stitched together for fashion (Figure 3.8-thing ID: 255924) and elastic textiles [90]. Servos were used to actuate a

mini quadruped robot (Figure 3.1f) and convert 3D models into robotic arms [61]. Second, kinetic objects were created to achieve similar behaviors. For example, by using embedded 3D-printed hinges, makers created transformable fidgets (*e.g.*, Figure 3.7-2595224) while researchers converted 3D bodies into transformable characters [136]. Wrapped conductive coils were powered to produce rotary motions in the maker community (Figure 3.1g) and research [85]. Finally, we saw similar actuation methods to trigger movements such as SMA (thing ID: 1917236 vs. [45]) and pressure (thing ID: 277171 vs. [68]). From the collected set of kinetic designs, springs stand out because they could be used as 3D printable actuators in many applications, such as a motor for a 3D-printed wind-up boat (thing: 3554489) and an extremely complex but self-functional gyrotourbillon (thing: 2820444). Besides above commonalities, a few creations on Thingiverse reused the exact techniques introduced by fabrication research. For example, the multi-material flexible plier (Figure 3.5-thing: 2433894) reused the metamaterial design from [41]. As some makers have explored making 3D-printed springs, it is timely to study springs for 3D printing and explore how 3D-printed springs could advance the making of kinetic objects for interactions.

While many commonalities are shared, a gap exists between the maker community and fabrication research. First, environmental stimuli were rarely used in practice for actuating kinetic behaviors, except for a few found examples (*e.g.*, Figure 3.1e). We speculate that such approaches were still at a research stage and not accessible to the public; hence, makers might seek other alternative methods. Second, novel applications such as fabric layered soft devices [86] are enabled by specialized fabrication machines and processes in research; however, due to the lack of access to those techniques, makers' creations were primarily limited by commercial 3D printing technologies. Lastly, the most significant missing part in the maker community was the design support for both parametric design and behavior control. While custom design tools were developed to allow the end-user to parameterize the model and preview the output behavior in fabrication research [41, 145], we saw limited design aids were provided for makers during the design and fabrication of kinetic objects—perhaps because these features were not integrated into existing CAD software. As a result, it was hard to tweak the created kinetic designs and control the resulting behavior. For example, the cellular structures in the metamaterial-based plier were fixed and hard

to extend to a different design without the support for parameterizing those grids. These gaps reveal several existing challenges for creating 3D printable kinetic objects: (i) designing kinematic models is complex and CAD support is missing for designers and makers to create functional, kinematic 3D designs; (ii) it is hard to achieve desired kinetic behaviors with appropriate mechanisms or materials without knowing how those components function; and (iii) it is demanding to print specific parts for kinetic behaviors, *e.g.*, tweaks on printing settings or specialized fabrication machines are needed, and error-prone post-print manual assembly is needed. As a fabrication researcher, I am passionate about exploring solutions to address these open challenges.

This study also introduces design opportunities for making kinetic objects using one of the less commonly used mechanisms—springs—for 3D printing. Like joints, gears, and microstructures, it is possible to create 3D printable springs and embed them in applications for movement and deformation. First, springs have been proved to be 3D printable (*e.g.*, thing: 3870039, thing: 171505, and thing: 746542) by makers, it is worthwhile studying how 3D-printed springs perform and how to control 3D-printed springs for expressive deformation behaviors. For example, Figure 3.6-thing:7760 shows that a spring suspension only extends and compresses with an internal linear guide. Amongst all the 3D printable functional mechanisms, 3D printable springs were rarely used in kinetic 3D designs due to the high barriers to designing and 3D printing springs. Most makers could only create basic spring forms, such as the standard helical or spiral springs, or reuse existing spring models for printing. Additionally, the substantial use of springs as motors in toys, such as catapults (*e.g.*, thing: 3662245), cars (*e.g.*, thing: 3308710), boats (*e.g.*, thing: 3554489), and clocks (*e.g.*, thing: 3061474), indicates that 3D-printed springs have the potential of being used as self-contained energy sources to trigger the movement of 3D kinetic objects. Hence, it has value to explore how to design and control 3D printable springs as motors to produce motion for 3D-printed objects in a broader range of applications. Lastly, since beam springs are widely used in many applications for expressive deformation behaviors, such as bending, rotating, and compression, it is possible to combine various beam structures to create interfaces that deform in a desired way and display controllable haptic feedback, such as the spring that produces different compliance (thing: 5713) and the ortho-planar spring for

larger deflections (thing: 3007261).

From this study, I drew the following design requirements for creating spring-based 3D printed objects that inform and guide the series of works in my dissertation: (i) springs need to be appropriately embedded in the custom 3D model for the ease of printing; (ii) spring forms can be customized for controllable deformation behaviors; (iii) springs can be controlled for the desired amount of energy to store when they are used as energy sources; And (iv) design aids should be provided for end-users who lack the engineering background to create spring-based 3D models for printing.

3.4 Chapter Summary

In this chapter, I presented *Making Things Move*, a large-scale study on how makers create 3D printable kinetic designs on Thingiverse. Through a qualitative analysis, I characterized the current trends in making 3D printable kinetic objects, including the kinetic mechanisms, output behaviors, actuation methods, and the design and making processes. While our findings and discussion highlight a set of identified creation patterns on Thingiverse, I also discussed the commonalities and differences between the maker community and fabrication research for making kinetic objects. The challenges for making 3D printable kinetic objects were enumerated. Finally, I discussed the opportunities for using springs as a new fundamental mechanism to create 3D printing kinetic behaviors. I described a set of design requirements that provide support for my dissertation.

Chapter 4

ONDULÉ: CREATING DEFORMATION BEHAVIORS WITH 3D PRINTABLE SPRINGS

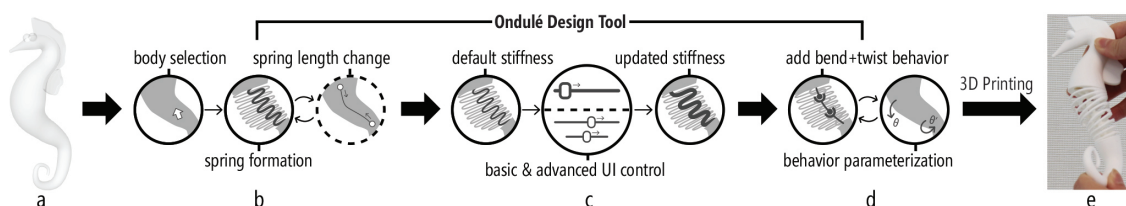


Figure 4.1: We introduce *Ondulé*, an interactive tool that allows designers to create and control deformable objects with embedded springs and joints. Above, a workflow shows how to make a solid seahorse body bendable and twistable: (a) select a seahorse body; (b) change the spring length and regenerate the spring directly on the model; (c) control spring stiffness; (d) parameterize spring deformation behaviors by adding additional joints; and (e) print the deformable seahorse with a consumer-grade FDM 3D printer.

To explore how springs can be used to make 3D printable kinetic objects, I started by studying helical springs for 3D printing and developed *Ondulé* [32] to lower barriers (RQ2) for the end-user to embed parameterizable helical springs in 3D models for 3D printable deformation behaviors (RQ1)¹. Compared to other mechanisms such as hinges, joints, or metamaterials, helical springs offer several benefits. First, they support various deformation behaviors, including compressing, stretching, bending, and twisting [11]. Second, because springs can endure large-scale deformations and store energy [11], they are ideal mechanisms for producing the driving force for motion. Finally, high-density spring structures have small gaps between each coil, making it possible to create complex deformable surfaces with a standard FDM 3D printer. While helical springs have compelling potential, two key challenges prevent them from being widely used in 3D printing. First, due to the anisotropic characteristic of additive manufacturing, the performance and mechanical properties of 3D-

¹The full video demo of *Ondulé*: <https://youtu.be/Zln1WlrDQ-4>

printed helical springs have not been extensively studied [23]. Second, although helical springs support a wide range of deformations, designing, customizing, and controlling the deformation is complex.

This chapter describes how I developed *Ondulé* to address these challenges. First, I provided the background on the physics of mechanical helical springs. I then present a series of controlled mechanical experiments that investigated the mechanical properties of 3D-printed helical springs. The results indicate that 3D-printed springs perform similarly to theoretical predictions. Next, to control the spring deformation behaviors, I presented a set of joint designs (*prismatic joints, revolute joints, knuckle joints*) embedded inside a 3D-printed spring, which can be used to customize and parameterize spring behaviors. Based on our experimental findings and custom joint designs, I developed *Ondulé*—a new interactive design plugin for Rhino that allows novices and makers to rapidly prototype 3D deformable behaviors with spring structures. With *Ondulé*, a user can select arbitrary shapes (Figure 4.1a), convert them to springs (Figure 4.1b), control and customize the desired deformation by parameterizing the spring and its internal joints (Figure 4.1c, 4.1d), and print the deformable object with a consumer-grade FDM 3D printer (Figure 4.1e). *Ondulé* also provides real-time feedback about the spring’s behavior (*e.g.*, its full compression position); however, a full dynamic simulation remains open work. Finally, to highlight the potential of *Ondulé*, I presented a series of 3D-printed applications designed with the tool.

This chapter contributes: (i) a set of novel spring deformation techniques with intrinsic mechanical joints that allow for spring behavior customization; (ii) an interactive design tool that allows designers to rapidly convert a static 3D model to a deformable and printable object by controlling spring stiffness and parameterizing additional joints; and (iii) a series of example applications created with *Ondulé* demonstrating the feasibility and initial design space.

4.1 Helical Spring Theory

Our approach is based on helical springs [11], which have three basic configurations—compression, extension, and torsion (p. 626 in [17]). Helical spring behaviors (Figure 4.2) are determined by two interrelated factors: spring parameters and material properties. We use both in our

tool.

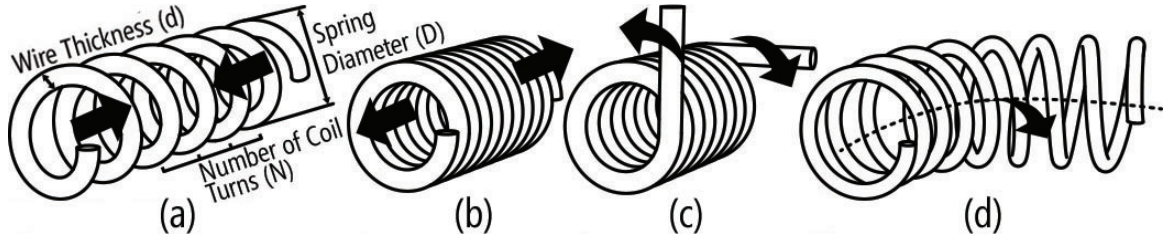


Figure 4.2: Basic helical spring deformation behaviors: (a) compress, (b) extend, (c) twist, and (d) laterally bend.

There are two primary factors that influence the mechanical performance of helical springs. **Spring parameters.** The compression and extension behaviors of helical springs can be modeled using *Hooke's Law* (Eq. 4.1) and *Castigliano's theorem* (p. 502 in [11]), where a spring's stiffness k is determined by wire thickness d , diameter D , number of coil turns N , and shear modulus G . Similarly, to model the torsion (*i.e.*, twisting) behavior of helical springs, we use the angular form of *Hooke's Law* (Eq. 4.2) and *Castigliano's theorem* (p. 534-535 in [11]), where a spring's torsion rate k' is determined by spring parameters and Young's modulus E . Given that a material's properties are constant, we can manipulate d , D , and N in our design tool to parameterize spring behaviors. **Material properties.** There are two relevant material properties to control a helical spring's behavior: Young's modulus (E) and shear modulus (G). (See Appendix B for the formal definitions of E and G and how we derived G). While E is typically listed in filament datasheets, this E is for a single, unextruded portion of filament, and G is not listed. Thus, to obtain these values, we need to measure them experimentally for each filament type (*e.g.*, ABS, PLA). We do so for one filament type below.

$$k = \frac{F}{x} = \frac{d^4 G}{8D^3 N} \quad (4.1)$$

$$k' = \frac{\tau}{\theta} = \frac{d^4 E}{64DN} \quad (4.2)$$

4.2 Mechanical Experiments

The mechanical experiments have two primary goals: first, to study the effect of 3D printing on the material properties E and G for our selected filament type; and second, to explore whether 3D-printed helical springs perform similarly to theoretical predictions. Towards the first goal, I conducted material property tests (Experiment 1) using a load frame to empirically measure E and G (Figure 4.3a), based on American Society for Testing and Materials (ASTM) standards. For the second goal, I evaluated the tensile (Experiment 2) and twisting (Experiment 3) performance of 3D-printed helical springs using a load frame and a torque sensor, respectively (Figure 4.3b and 4.3c). Experimental results are used to inform *Ondulé*'s parameter space and for the spring preview. The experiments focus solely on the performance of 3D-printed helical springs and do not include joints.

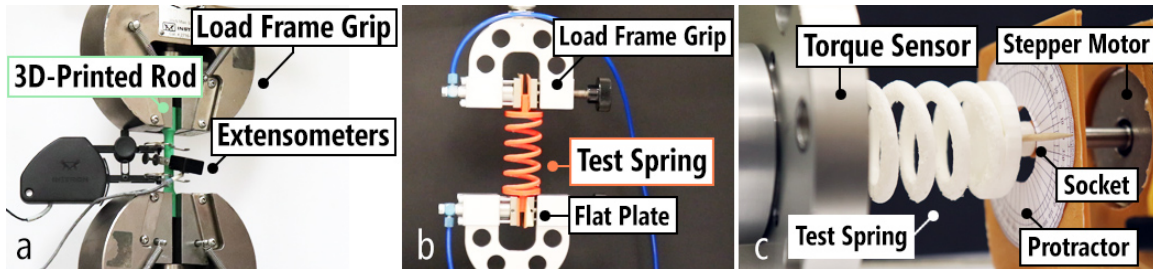


Figure 4.3: Mechanical experiment setups: (a) the load frame stretches a 3D-printed rod; (b) the load frame stretches a 3D-printed helical; and (c) the motor rotates a helical spring and torque is measured.

All test samples were printed with tough PLA (TPLA) and dissolvable PVA using an Ultimaker 3 printer. PVA was used for support material and fully removed before the experiments. I used Ultimaker Cura 3.6.0 with default print settings and varied infill density, infill pattern, and printing orientation (depending on the experiment). Our experiments were run under the supervision of and consultation with a mechanical testing lab engineer.

4.2.1 Experiment 1: 3D-Printed Rod Material Property Tests

To derive E and G for 3D-printed TPLA and explore the effect of 3D printing on these properties, we directly measured E and the Poisson ratio ν for different 3D printer settings (Figure 4.4). The experiment setup is shown in Figure 4.3a and detailed in Appendix B.

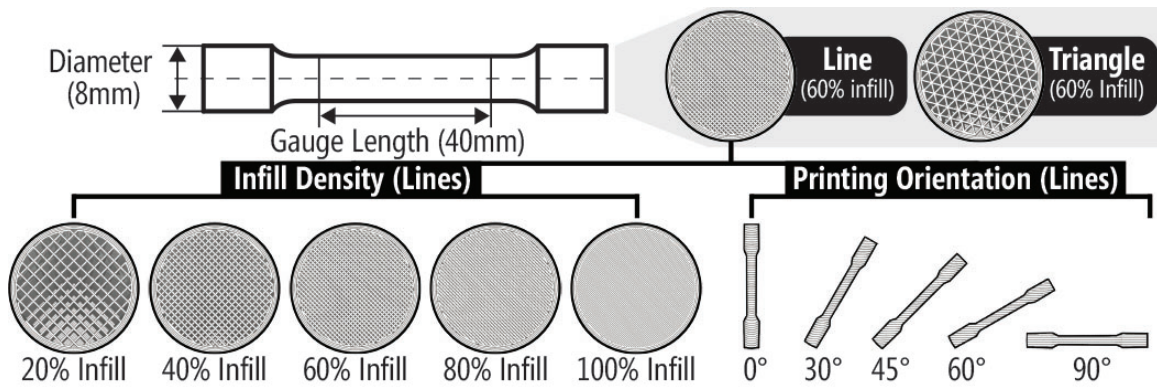


Figure 4.4: The 3D-printed solid rods in Experiment 1 and three varied test conditions: infill density, infill pattern, and print orientation.

The experimental results show that (i) stiffness increases as infill density increases, (ii) tensile strength orthogonal to the printing direction is highest, and (iii) shear stress is highest at a 45° angle (Figure 4.5). See explanations of the results in Figure 4.5 in Appendix B.

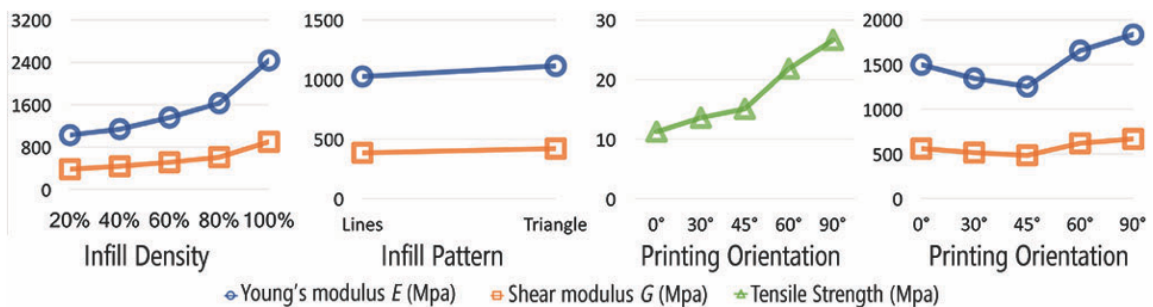


Figure 4.5: Experiment 1 results showing that E and G increase with infill density as well as more robust infill patterns. Tensile strength increases as printing angle increases; however, shear stress is highest at 45°.

Table 4.1: The conditions of spring parameters used in Experiment 2 and Experiment 3.

Condition	Wire Thickness (mm)	Diameter (mm)	Length (mm)	Turn Number
Wire Thickness	2, 3.4, 4.8, 6.2, 7.6	25, 30, 50, 60	50	5
Diameter	4	32	25, 45, 65, 85	5
Spring Length	4	32	50	5
Turn Number	4	32	50	4, 6, 8, 10

4.2.2 Experiment 2: 3D-Printed Spring Tensile Tests

To empirically explore how the tensile performance of 3D-printed springs compares to theoretical predictions, I conducted controlled stretching experiments again using the load frame. I varied four spring parameters: wire thickness d , diameter D , the number of coil turns N , and spring length L (Table 4.1). While spring theory [11] suggests that length L has no effect on tensile performance, I also varied this parameter for verification. In all, I created and tested 17 helical springs with the same FDM specifications: 100% infill, lines infill pattern, and 90° printing angle. For the experiments, I followed a similar procedure to Experiment 1 (see the setup in Figure 4.3b and Appendix B). If our experimental results find that 3D-printed helical springs behave similarly to theoretical predictions, I can then operationalize spring theory in the *Ondulé* tool (*e.g.*, by allowing the user to control thickness d , diameter D , and the number of coil turns N).

To investigate how a 3D-printed spring behaves, I compared the empirically measured spring stiffness k of each spring to a k derived from Eq. 4.2. For d , D , and N , I simply use the experimental conditions (Table 4.1) as input values. For G , I used the value derived from Experiment 1 for 100% infill, lines infill pattern, and 90° printing angle. Using a paired (two-tailed) t-test, I found no significant difference (Figure 4.6) between the empirically measured k and the theoretical prediction ($t_{32} = 0.0097, p = 0.99$).

4.2.3 Experiment 3: 3D-Printed Spring Torsion Tests

Finally, I investigated the torsion (twisting) performance of 3D-printed springs for the last experiment. We reprinted the same springs used in Experiment 2 (Table 4.1) with the same

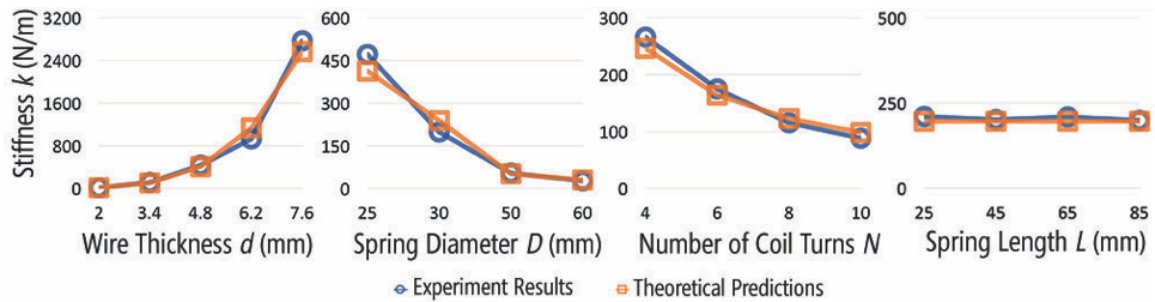


Figure 4.6: Experiment 2 results showing that 3D-printed helical springs perform similarly to theoretical predictions as measured by a load frame with different d , D , N , and L values.

FDM specifications (100% infill, lines infill pattern, and 90° printing angle); however, we used a different experimental setup (see Figure 4.3c and Appendix B).

Similar to Experiment 2, I compared our empirical results—in this case, the measured torsion rate k' for each spring—to theoretical predictions (Eq. 4.2). As before, I can input the experimental condition values for d , D , and N into Eq.4.2 as well as the material property E from Experiment 1 to derive the theoretical prediction k' for each spring. Figure 4.7 shows our measurement for k' closely mirrors the theoretical prediction based on a paired t-test ($t_{32} = 0.0236, p = 0.98$).

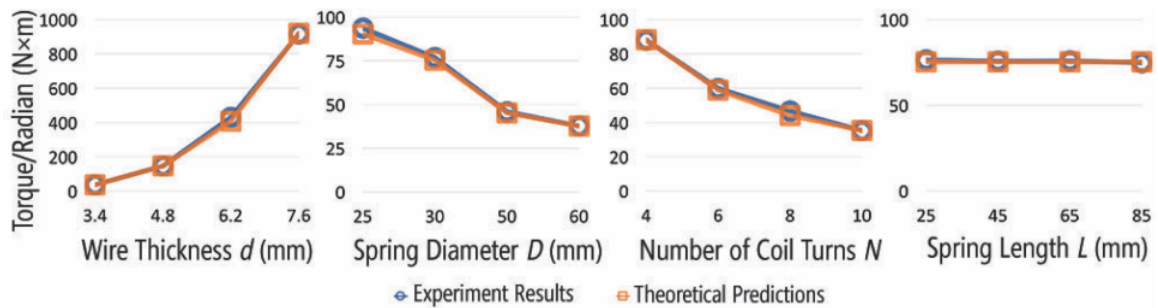


Figure 4.7: Experiment 3 results showing that 3D-printed helical springs have similar twist-performance to theoretical predictions with varied d , D , N , and L values.

4.3 Spring Deformation Techniques

The experiments derived E and G for 3D-printed TPLA demonstrated the feasibility of 3D printing helical springs using FDM 3D printers and showed that 3D-printed helical springs perform similarly to theoretical predictions. Informed by spring theory, the above experiments, and our own extensive testing of 3D-printed helical springs, I created custom spring deformation techniques, which use mechanical joints as constraints for enabling our approach. To enable users to create parameterizable springs, this set of deformation techniques combines auto-generated helical springs with embedded mechanical joints. The combination of springs and joints is also seen in our *MTM* study, for example, spring-loaded suspensions use internal linear guides to limit the spring to compress and extend only². In our designs, for some deformations, I also enable users to auto-generate a lock mechanism that allows the spring to be locked/unlocked in a fixed position (currently supported by *linear-only* and *twist-only* deformations).

4.3.1 Controlling Linear Deformations Using a Prismatic Joint

To create a spring that can only compress or extend, I embed a prismatic joint consisting of a shaft, a rail guide, and an embedded slider (Figure 4.8). When the spring is compressed or stretched, the slider can only move along the predefined rail, preventing the spring from being bent or twisted. The user can specify the amount of compression and/or extension, which I control by positioning the default location of the slider. Consequently, this joint design can be used to support *compression-only*, *extension-only*, or *both*. The user can also auto-generate a locking mechanism, which can be used to lock a spring at its maximum compression and/or extension states. I do this by generating a ‘latching’ groove at the endpoints of the guide rail (Figure 4.8).

4.3.2 Controlling Twisting Deformations Using a Revolute Joint

To create a spring with the *twist-only* behavior and to control the maximum angle of rotation, I embed a revolute joint using a bearing socket and a circular disc (Figure 4.9).

²Spring suspensions: <https://www.thingiverse.com/thing:3551> and <https://www.thingiverse.com/thing:7760>

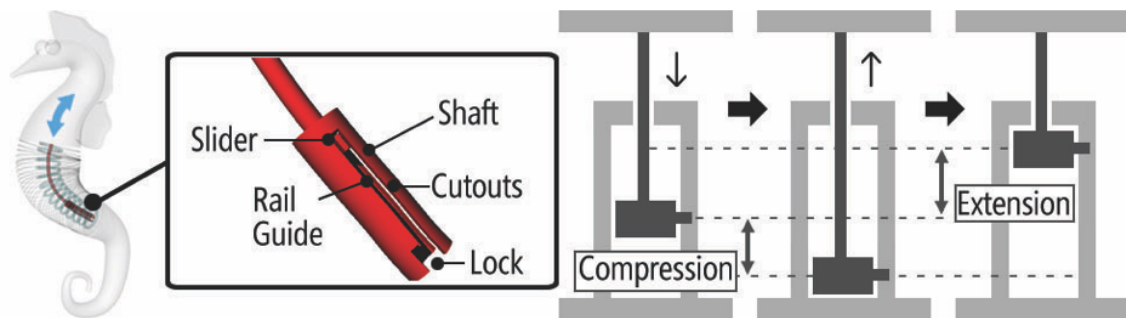


Figure 4.8: A prismatic joint is used for a linear-only deformation.

This revolute structure allows the circular disc to revolve concentrically to the bearing socket while preventing the bending of linear deformations. For controlling the angle of rotation, I embed a custom arc sliding rail in the socket, which confines the rotation of the disc within the maximum twisting angle. Again, I generate a ‘latching’ groove at the position where the disc rotates to the maximum twisting angle to lock a spring at its maximum angle.

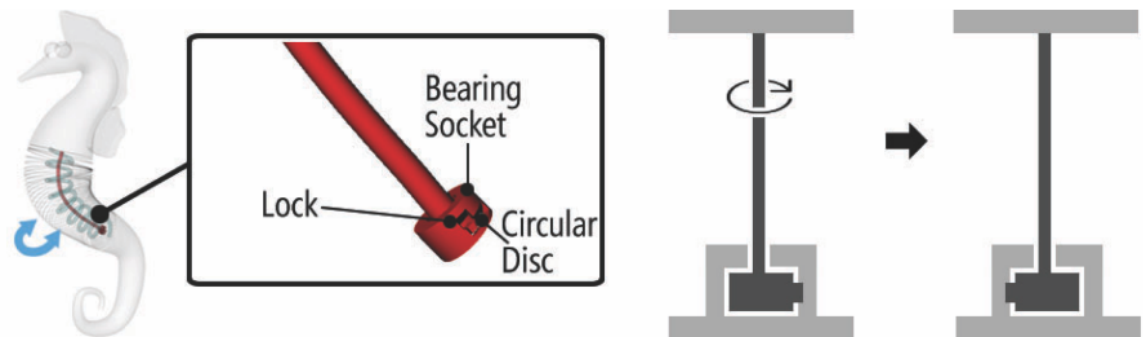


Figure 4.9: A revolute joint is used for a twist-only deformation.

4.3.3 Controlling Bending Deformations Using a Knuckle Joint

To support *bend-only* behaviors, I use a chain of knuckle joints. A single knuckle joint contains a cylindrical rod located inside a cylindrical socket (Figure 4.10). The cylindrical rod revolves concentric to the axis of a cylindrical socket. Multiple knuckle joints can be

chained nose to tail, with the first and the last one fixed to the two ends of the helical spring. This structure prevents linear and twisting deformations and offers flexible bending. However, knuckle joints confine the bending deformation to one plane. Therefore, an omni-bend-only deformation is also possible, which I describe below.

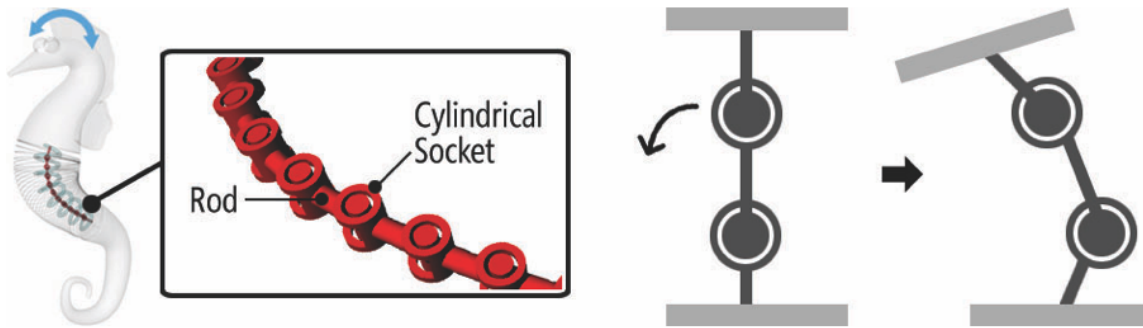


Figure 4.10: A chain of knuckle joints used for bend-only deformations.

4.3.4 Supporting Compound Behaviors

While the above techniques support individual deformation behaviors, compound behaviors are also supported in three ways: first, via traditional freeform springs (unconstrained by internal joints), second, via additional joint designs, and finally, by combining multiple springs in serial or parallel.

Free-form springs. Users can select geometries and convert them to free-form springs (without embedded joints). These springs can inherently *compress*, *extend*, *bend*, and *twist*. Here, users can only control spring stiffness (via parameters d , D , N , and L) and shape.

Additional joint designs. Using custom joint designs, I support two compound behaviors: *linear+twist* and *twist+bend*. For *linear+twist*, I adapt the linear-only design by replacing the slider with a circular disc in the shaft, which is a *cylindrical joint* (Figure 4.11a). This enables the disc to glide and twist along the rail guide. For *twist+bend*, rather than embedding a chain of knuckle joints, I embed a chain of ball joints. This allows the spring to twist and bend at any angle (Figure 4.11b).

Serial/parallel combinations. Multiple springs can be combined in serial or parallel

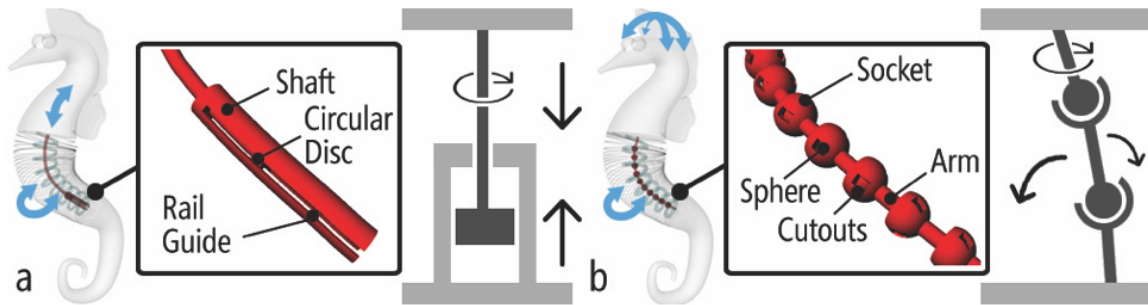


Figure 4.11: (a) A cylindrical joint is used for linear+twist deformations and (b) a chain of ball joints is used for twist+bend deformations.

to further produce more complex deformations. For example, the snake design in Figure 4.21 combines different spring types serially; the hand exerciser (Figure 4.19) uses *linear-only* springs in parallel.

4.4 Interactive Spring Design Tool

The above techniques are integrated into our custom interactive design tool, *Ondulé*, via a plugin for Rhino 5. *Ondulé* enables novices to rapidly create deformable 3D-printed objects using embedded springs. To use *Ondulé*, the user (1) models geometries (i.e., 3D bodies) in the traditional Rhino CAD environment; (2) selects specific bodies and converts them to springs using *Ondulé*; (3) then parameterizes spring stiffness; (4) and specifies the spring deformation behavior (e.g., linear-only or twist). Stages 2-4 are supported via a side panel in Rhino (Figure 4.12). Below, I describe stages 2-4 before providing details on the implementation and underlying algorithms.

4.4.1 Generating Springs

After creating a 3D model in Rhino, the user can use our tool to generate spring structures. To generate a spring, the user selects a 3D body in Rhino and then clicks on the ‘*Convert to spring*’ button. If the selected body is cylindrical with a consistent diameter, the 3D shape is automatically converted into a deformation spring. If the selected body is determined to be a non-cylinder geometry, our tool will convert it into two springs: an internal deformation

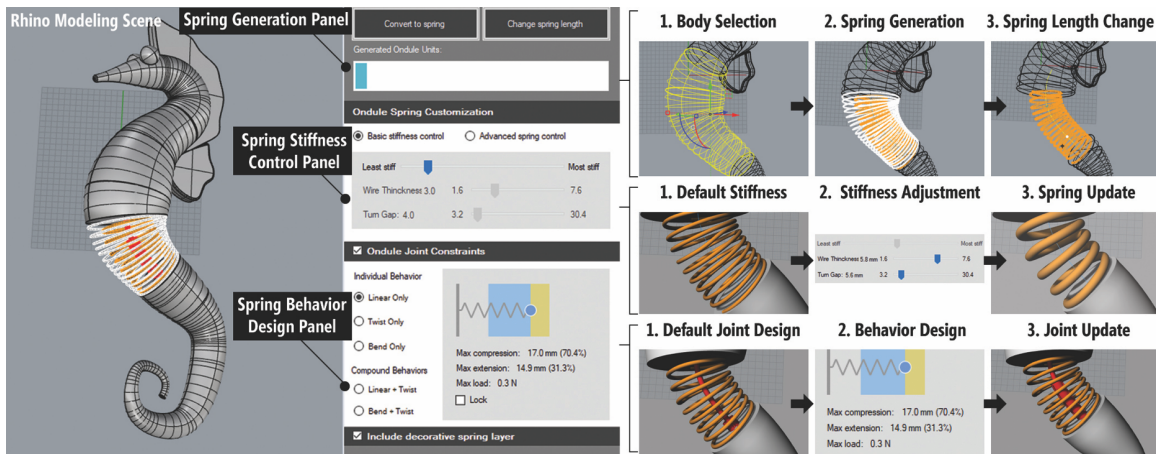


Figure 4.12: The *Ondulé* spring design tool interface (left) has four parts: Rhino modeling environment, a spring generation panel, a spring stiffness control panel, and a spring behavior design panel. The workflow for each design panel is shown on the right.

spring that follows the medial axis of the selected geometry (Figure 4.12) as in the cylindrical case and an extra outer decorative spring that follows the body’s geometric form and is created with dense and thin layers of coils ($d=1.6\text{mm}$), which has a minimal effect on the overall stiffness. The decorative spring maintains the complex topology of the selected geometry, while the internal spring serves as the functional spring for deformations. The decorative spring (if generated) is default hidden to reduce visual clutter, though it can be turned on via a checkbox.

4.4.2 Controlling Spring Stiffness

The system automatically generates the spring diameter D and its length by default, but these parameters can be adjusted. The users can change the spring stiffness either using a simple slider or by directly modifying the spring wire thickness d and the number of turns N (Figure 4.12).

4.4.3 Specifying Deformation Behavior

Finally, the user can specify the spring’s deformation behavior: *linear-only*, *twist-only*, *bend-only*, *linear+twist*, or *twist+bend*. For each deformation, we provide a custom UI panel. For

the compound behaviors, we combine the UI from their respective individual panels.

Linear-only. For *linear-only*, the user specifies the *maximum compression* and *extension* points of the spring. We provide real-time feedback about the spring’s displacement (shown in millimeters and percentage of L) as well as the estimated force (in Newtons) required for that displacement. The user can also click on the ‘*Lock*’ checkbox to auto-generate a lock mechanism at the maximum compression and extension points.

Twist-only. For *twist-only*, the user can control the *maximum twisting angle* up to 90° , which we found is a safe maximum angle preventing the spring from buckling in our torsion test. As the user drags the angle selector, the UI shows the selected angle (in degrees) as well as the estimated force (in Newtons) required to reach that angle. Like the linear-only UI panel, the user can add a lock mechanism at the maximum twisting point by clicking on the ‘*Lock*’ checkbox.

Bend-only. For *bend-only*, the user first specifies the *bending direction* via an angle selector, which overlays a 3D direction indicator on top of the model in Rhino, and then specifies the maximum *bending angle* using a second angle selector (shown in degrees). Unlike the other UI panels, the tool does not show force estimates. As noted in the Theory, modeling bending behavior is an open area of research.

4.5 Implementation

Ondulé was implemented in C# using Rhino 5’s plugin architecture (RhinoCommon API³). Below, we describe how we computationally generate the springs, the embedded joints, and the locking mechanism.

4.5.1 Springs Generation

To generate deformation springs, the tool first computes the medial axis of the selected 3D shape using a *mean curvature flow* (MCF) algorithm. The medial axis is a skeletal curve at the center of the selected 3D body. It is used to evaluate if a spring structure can be successfully converted and if a decorative spring is needed to preserve the 3D shape’s

³RhinoCommon API: <https://developer.rhino3d.com/api/>

appearance.

To determine printability, the tool computes the average minimal distance dis_{crv} from the medial axis curve to the 3D shape surface using fixed sampling (Figure 4.13). If the distance is larger than the minimum diameter of a printable spring (3.6mm for D based on our mechanical experiments) the selected shape can be converted to the spring structure. Using dis_{crv} the tool also evaluates if the selected shape has a complex surface topology that is worth preserving with a decorative spring. This is done by comparing the actual variance of the samples with respect to the mean. If the variance is above a certain threshold, the tool generates the decorative spring.

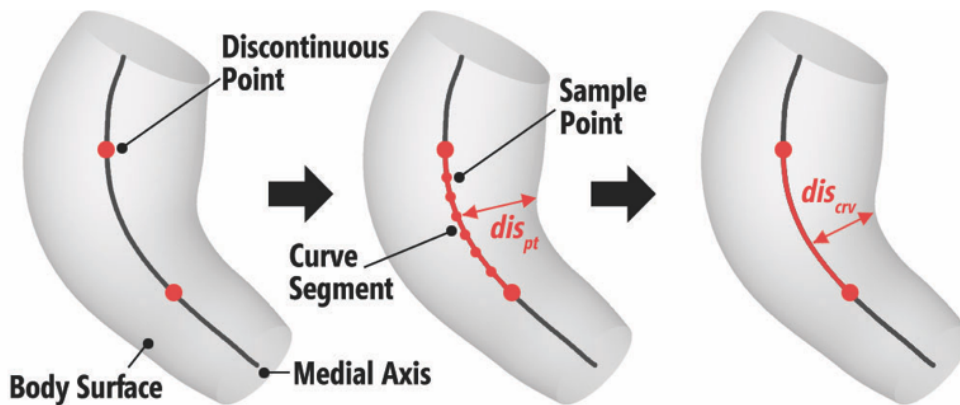


Figure 4.13: Generating the medial axis, calculating the size of the selected body, and evaluating the printability of an embedded spring.

To generate the deformation spring (Figure 4.14), the tool uses the *RhinoCommon spiral* function. This function produces a spiral curve following the medial axis. The spiral curve is then used as a parameter for the *sweep* function to create the final springs. Finally, all spiral solids are concatenated to construct a spring (highlighted in yellow in Figure 4.14).

Finally, if needed, the tool generates a decorative spring to preserve the original 3D shape appearance. Note that the *spiral* function cannot be used directly for the decorative spring generation, as it can only produce a helix with a consistent radius. Instead, the tool first reuses the spiral curve generated for the deformation spring described above and projects it onto the 3D shape surface using a 300-sampling point (Figure 4.15). The tool

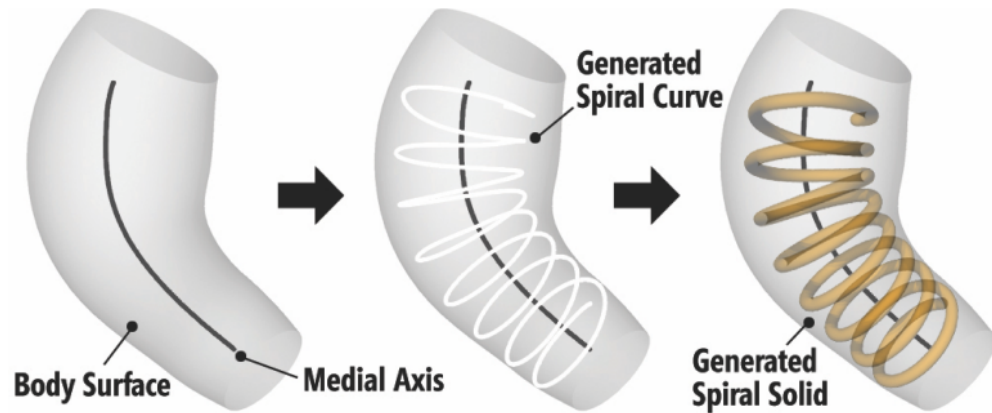


Figure 4.14: Generating the deformation spring using the generated medial axis and RhinoCommon functions.

then generates a new set of points by retracing all intersecting points toward the median axis curve by a fixed decorative spring wire thickness (*i.e.*, 1.6mm) and creates a new *curve* object by interpolating these points. Finally, using this curve, the *sweep* function is applied to create the final decorative spring.

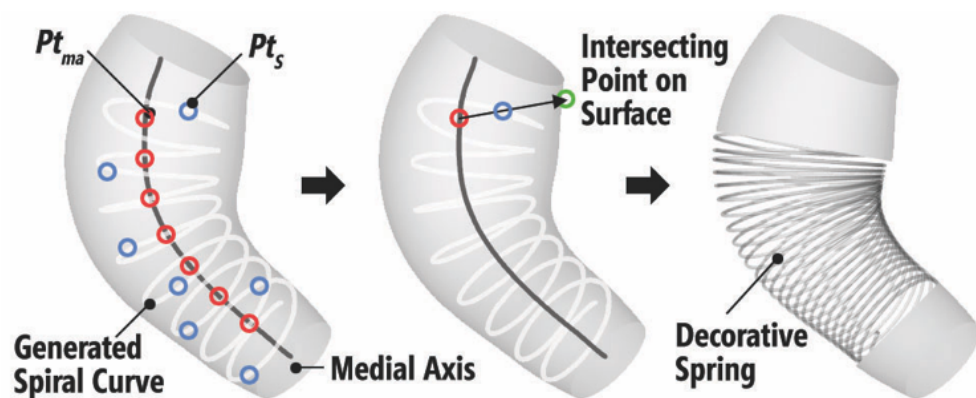


Figure 4.15: Generating the decorative spring.

4.5.2 Generating Embedded Joints

The tool computes the embedded joints using the medial axis generated from the previous steps. For *linear-only* joint, the tool first decides the slider's starting position on the medial axis and calculates its possible extension and compression distances based on the user's input. The tool then extrudes the slider rod, the shaft, and the rail guide sweeping along the medial axis (Figure 4.16a). *Twist-only* and *linear+twist* joints share a similar procedure, except that the slider position varies in joint design.

Bend-only and *twist+bend* behaviors use chained joints design. To generate these joints, the tool first calculates the number of joints needed in the selected body. Then, for each joint (*i.e.*, a knuckle joint or a ball joint), the tool determines the position and the length of the inner bearing stud on the medial axis. Next, the tool extrudes the solid stud along the medial axis and generates a cylinder (for *bend-only*) or a sphere (for *twist+bend*) at the endpoint of the stud. Finally, the tool generates the outer bearing socket for each joint and then connects it to one end of the next joint's bearing stud, resulting in a chain structure. Figure 4.16b shows an exploded view of the construction of the ball joint chain.

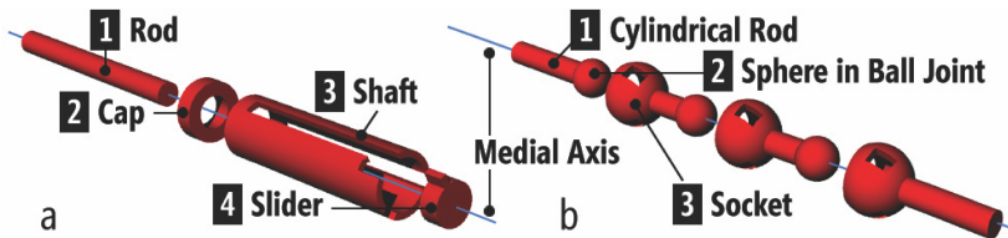


Figure 4.16: An exploded view of (a) a prismatic joint and (b) a chain of ball joints, which are used for *linear-only* and *twist+bend* respectively.

4.5.3 Generating Locks

The tool uses latching grooves as the locking mechanisms for *linear-only* and *twist-only* deformation behaviors. For the locking mechanism in *linear-only* design, the tool first locates the endpoints of the shaft and then generates a groove next to the rail guide by

executing a Boolean difference operation from the shaft with a *cubic* object. To avoid the slider from slipping out of the groove, the tool also generates a fence on the edge of the groove (Figure 4.8). Similarly, for the revolute joint in *twist-only* design, the tool generates the groove at the position where the disc rotates at the maximum twisting angle (Figure 4.9).

4.6 Validation through Applications

To evaluate *Ondulé* and highlight an initial design space, I created five examples, each emphasizing one or more features of *Ondulé*.

4.6.1 Jack-in-the-box

The jack-in-the-box is one of the most well-known helical spring-based toys. Here, I showcase how conventional spring-based mechanisms can be enhanced using the *Ondulé* design tool. Figure 4.17 showcases the jack-in-the-box design, where instead of using an unconstrained helical spring, the center spring is built to follow a predefined curve and with a compress+twist deformation behavior, which is achieved by using a freeform spring and a spring with a cylindrical joint. After cranking, a cat figure pops out of the box with a turning motion. The spring and the cranking components were 3D printed, and the box was made by laser cutting.

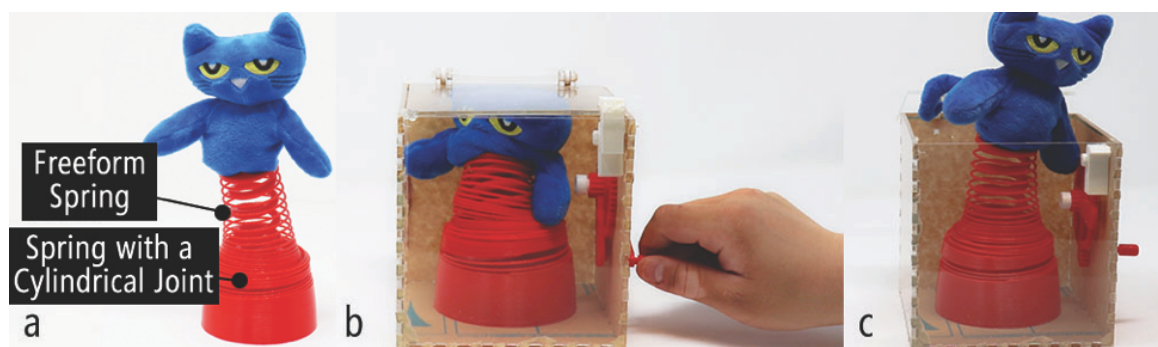


Figure 4.17: A jack-in-the-box spring mechanism generated by *Ondulé*: (a) two spring designs are embedded; (b) the cat can be fully compressed and locked inside a laser-cut box; and (c) the cat pops out following a path as a surprise.

4.6.2 Launching Rocket

Another set of spring-based applications uses the deformation to produce the driving force for mechanical motion, such as the proper shooting motion in our launching rocket example (Figure 4.18). I first select the smoke shape and convert it into a compress-only spring by adding a prismatic joint. I then created an additional latch structure, which when released will push the rocket to fly straight up. Note that though a simple example, such an application will be difficult to create without *Ondulé*. First, by adding a prismatic joint, we can ensure that the rocket will be pushed in the right direction without twisting or bending during the launching. Second, the prismatic joint resides inside the conical smoke shape, serving the linear constraint function with minimal effect on the 3D shape appearance.



Figure 4.18: A launching rocket application: (a) a rocket sits on top of a compressed “smoke” spring, which is locked by an external latch; (b) the user can launch the rocket by pulling the latch; and (c) the smoke is in its full extension.

4.6.3 Hand Exerciser

The previous two examples showcase how I can use *Ondulé* for single spring deformation behavior. Here, I demonstrate how multiple springs can be created and customized through a set of hand exercisers (Figure 4.19). Commercial hand exercisers are widely used for hand rehabilitation or Arthritis therapy. Figure 4.19a is a commercial design with multiple springs in parallel for finger exercises. Figure 4.19b is our *Ondulé* replication using four freeform springs in parallel for each finger and three extra springs for the palm. One limitation of the commercial hand exercisers is that springs for all the fingers have the same stiffness. As

such, it would be impossible for a user to exercise different fingers with different strengths. To address this limitation, I designed a custom hand exerciser (Figure 4.19c) in which, for each finger, the user can customize the spring stiffness by adjusting the spring parameters d and N . Finally, users can also create their own designs in arbitrary geometries, for example, a blowfish shape (Figure 4.19d).

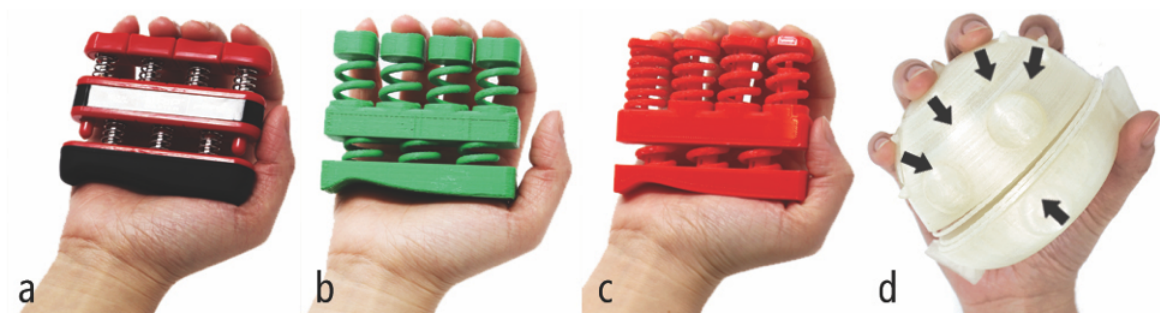


Figure 4.19: Replicated and custom hand exercisers: (a) an off-the-shelf hand exerciser (b) a replication with 3D-printed springs, (c) a custom design that includes springs with different stiffnesses and custom prismatic joints for compress-only behavior, and (d) a blowfish version.

4.6.4 Tangible Prop for Storytelling Authoring

Custom 3D-printed deformable springs can be further combined with external electrical components to create expressive interactions. In this example, I created a digital storytelling authoring tool with a tangible prop made with a 3D-printed animal and low-cost sensors (Figure 4.20).

The 3D-printed prop comprises a stretchable neck (with a prismatic joint design and a lock mechanism), a bendable body (with a knuckle joint design), and four freely deformable legs. With only one physical design, two animal characters (*i.e.*, horse and giraffe) with two separate actions (*i.e.*, talking and walking) can be mapped to the prop due to the spring design. For example, a digital horse will appear in our authoring tool when the prop’s neck part stays unstretched (Figure 4.20b). When extended, the physical prop has a longer neck thus a giraffe will show up on the screen accordingly (Figure 4.20c). To detect the length change of the neck, a linear hall effect sensor is used and attached to the bottom of the neck

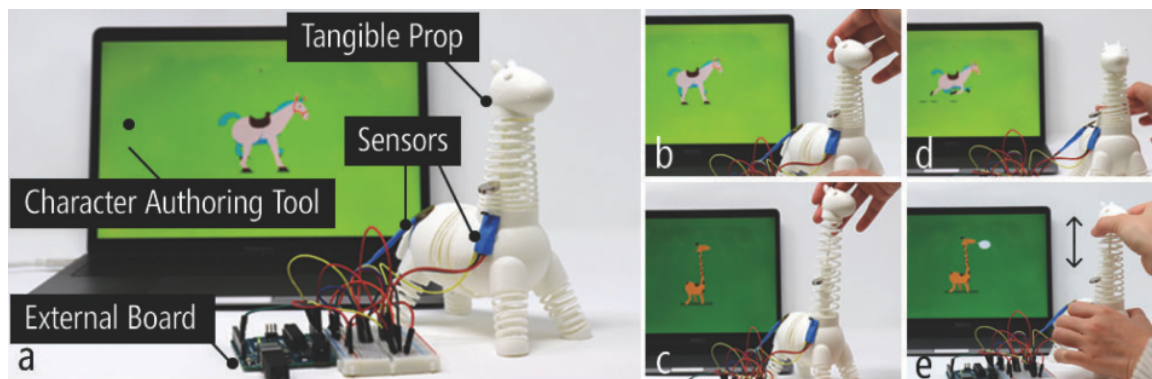


Figure 4.20: The setup of a tangible storytelling prop.

with a magnet fixed above it (Figure 4.20a). The sensor is connected to an Arduino, which communicates to the authoring tool developed with Processing. While either character is activated, moving the head up and down can trigger the character’s talking action (Figure 4.20e). I can also attach a piezo sensor to the prop body to detect vibration, which can be used as a walking action trigger (Figure 4.20d). The sensor can robustly detect the repetitive tap due to the converted bendable body and springy legs.

4.6.5 Other Applications

Here I showcase other applications designed using *Ondulé*. I collaborated with a mechanical engineering team and created an accessible cutting device for people with muscle weakness (Figure 4.21a). With the *Ondulé* design tool, we could rapidly make custom springs that could fit in the cutting device and offered the exact amount of stiffness needed for the patient. Figure 4.21b shows a Halloween mask as an example of wearable fashion. The elephant’s trunk was designed with a chain of ball joints so that it was flexible, bendable, and lightweight to wear. Finally, various joints could be incorporated into a snake body in a serial arrangement (Figure 4.21c). I envision a robotic snake controlled by an external control system and actuators.

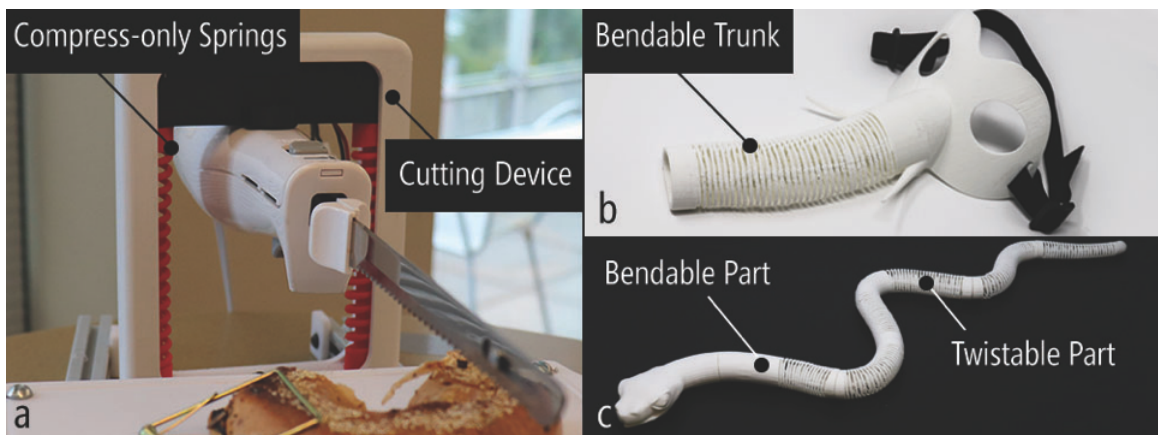


Figure 4.21: Other applications that *Ondulé* can support: (a) an accessible cutting device, (b) an elephant mask with a bendable trunk, and (c) a snake body with multiple spring deformation behaviors.

4.7 Limitations

The main objective of *Ondulé* is to support deformation customization with 3D-printed springs. I achieve this by inserting custom joints inside helical springs, so spring behaviors become modular without requiring external supporting structures. However, this approach also has several limitations.

Geometry Printability. The size of an object is limited to the minimum joint size that we can print (currently a minimum 0.4mm tolerance is needed between the moving and the stationary parts of the joint) and the minimal spring wire thickness (1.6mm in our current setup) that is printable. As a result, it is difficult to convert geometries with smaller diameters (*i.e.*, smaller than 3.6mm) with our current tool. However, joint size and spring wire thickness could be further reduced with a higher resolution printer or alternative printing process (*e.g.*, SLA). Further, the current approach may affect the degree of deformation. For instance, compression with our current prismatic joint design cannot surpass half the length of the original spring. Similarly, a stretching behavior with the prismatic joint cannot exceed two times its original length, even though the helical spring solely could be further extended. One solution to this is to consider an alternative joint design. For instance, we can replace the rail with a prismatic telescoping structure [134].

Influence of Friction Force and Decorative Spring. The mechanical experiments focused solely on the behavior of 3D printed helical springs. With the embedded mechanical joints, I further understand its persistent friction force with an additional experiment. Here, I compared the oscillating behavior of two springs printed with and without a prismatic joint (*S1* and *S2*, respectively). We observed the relaxation behavior of these springs when stretched to the same length in three orientations (vertical—0°, 45°, and horizontal—90°). The results show that *S1* comes to rest 48%, 50%, and 70% faster, respectively. As expected, the horizontal setting has the most friction. These preliminary results indicate that friction force exists in 3D-printed joints, which impede normal spring behavior. Future work should explore methods to reduce friction. For example, we found that joint-based friction can be diminished by using a filament material with a lower friction coefficient (*e.g.*, ABS) or adding grease. We confirmed that an ABS spring reduces friction by 24% in the oscillation test setting. For some 3D models, *Ondulé* generates both a deformation spring and a decorative spring—the latter enables us to approximate organic surface topology while maintaining form. The decorative spring is not intended to influence overall spring behavior, only the aesthetic. To examine the effect of the decorative spring on spring performance, I compared the overall stiffness k of springs with and without an added decorative spring. As desired, we found a minimal impact: an increase of 0.02% in the jack-in-the-box application and 0.21% in the rocket application. When analyzing the current *Ondulé* parameter space, the expected impact could be up to 4.42%. These preliminary results indicate that the decorative spring has minimal effects on the overall deformation of a 3D-printed object.

Simulation for the Combined Behavior. *Ondulé* allows a user to design the spring+joint deformation with a preview of its starting and end positions. However, the tool cannot currently simulate combined deformation motions. In the future, I plan to provide a more realistic validation for the spring+joint behaviors by simulating the effect of the intrinsic spring weight (with specific printing settings), the friction force of the internal joints, and external forces (*e.g.*, the applied user interaction force). One approachable solution is to use a physical engine such as *Kangaroo*⁴, which can be integrated into Rhino to build an

⁴Kangaroo Physics: <https://www.food4rhino.com/en/app/kangaroo-physics>

interactive simulation environment that offers designers a realistic motion and mechanical preview.

Spring Robustness. As discussed in Mechanical Experiments, printing orientation will affect the spring’s E and G , where 45° results in minimum values for both and 90° yields the highest. As such, all models presented in this paper are printed with the spring perpendicular to the 3D printer’s Z -direction for robust printing results. However, when multiple springs with varied orientations need to be printed at once, finding an ideal printing orientation that works for all springs would be challenging. As the simplest solution to this problem, we can take a print-and-assembly approach, where certain springs can be printed separately with the optimal printing angle and then affixed to the original model. Another approach is to involve an alternative 3D printing method. The 5-DOF 3D printing method can be used in this case to always repose the 3D model to ensure the spring can be aligned with the Z -axis. Since the impact of orientations on strength can be characterized, a constant overall stiffness can be achieved.

4.8 Chapter Summary

In this chapter, I presented *Ondulé*, an interactive design tool that allows the user to rapidly design and build deformable plastic objects with parameterizable springs and mechanical joints. First, I provided the background about the physics of mechanical helical springs. To develop *Ondulé*, I started with a series of controlled mechanical experiments that studied the feasibility of 3D-printed springs. The results of these mechanical experiments confirmed that 3D-printed helical springs compress, extend, and twist as the theory predicts. I then describe a set of spring and joint designs informed by the results of the mechanical experiments. Those deformation design techniques enable the customization of a spring’s deformable behaviors. Next, to allow the end-users to use these spring and joint-based design techniques, I developed the design tool—*Ondulé* where novices can quickly add a wide range of spring structures and deformation behaviors to an existing model. Finally, I showcase a series of example applications to demonstrate the potential and breadth of this approach and discuss the limitations of the current approach as well.

Chapter 5

KINERGY: ENABLING 3D-PRINTED OBJECTS WITH SELF-PROPELLED MOTION USING SPRINGS

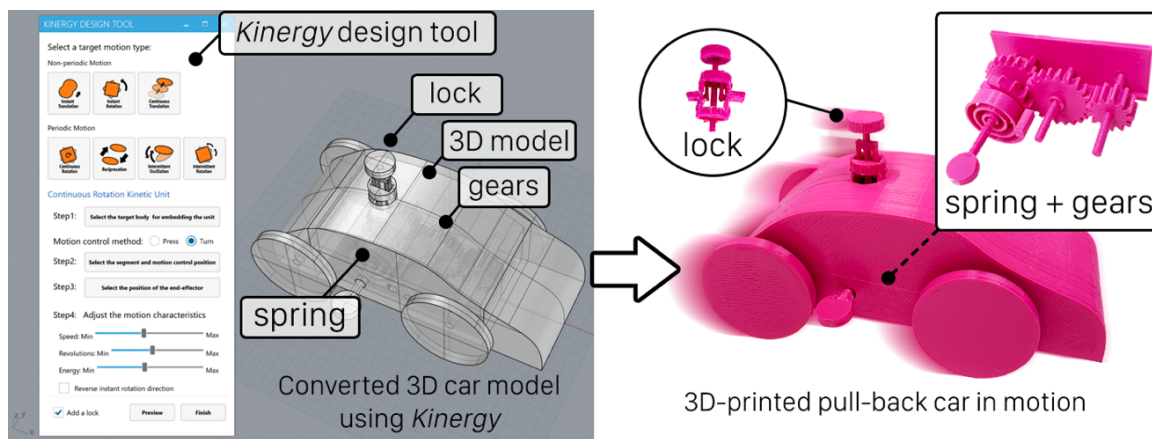


Figure 5.1: We introduce *Kinergy*, an interactive design tool to rapidly create 3D-printable energy-powered motion. Above, we show a 3D-printed pull-back car created with *Kinergy*: the static car model is converted into a motion-enabled model with an auto-generated and embedded spring, a spring lock, and a set of gears (left). All the parts in the converted 3D car model are printed in place with a commercial 3D printer and the printed car is ready to move without post-print part assembly (right).

While *Ondulé* investigates helical springs as the key elements to support 3D printable deformation behaviors, I further explored how to leverage the energy stored in springs for actuation. In the second project, I developed *Kinergy* [33]—an interactive design tool (RQ2) to enable designers to create 3D models with self-propelled motion (RQ1) by harnessing the energy stored in the embedded 3D printable springs¹. 3D-printable kinematics opens new exciting opportunities for 3D printing: designers can create and control printable movement enabling a wide variety of applications from 3D-printable rotational solar system models² to

¹The open-source code repository for *Kinergy*: <https://github.com/EdigaHe/Kinergy>

²<https://www.thingiverse.com/thing:3928677>

robotic quadrupeds ³. Despite substantial recent work on 3D-printable kinematic elements [13, 16, 145, 138] and actuation methods [14, 64], designing fully functional, 3D-printable kinematic objects with controllable movements remains challenging. Expert skills, time, and labor are needed to create the 3D kinematic models [38, 121], assemble multiple 3D-printed parts, and interface components with external power sources for actuation [45, 128, 108].

In this chapter, I describe how *Kinergy* addresses these challenges. First, my approach provides the background of motion types for 3D printing. Then I introduce a suite of custom, 3D printable mechanical structures called *kinetic units*, which are central to *Kinergy*'s approach. These kinetic units encapsulate complex kinematic and parametric mechanisms as "black box" units, abstracting the design and control of complex 3D motions into a direct manipulation interface. They integrate 3D printed helical or spiral springs as self-contained energy sources, convert spring deformations into controllable motion behaviors via mutually engaged kinematic transmission mechanisms (*e.g.*, gears, rack-n-pinions), and provide triggerable motion via embeddable compliant lock mechanisms. Kinetic units offer three key benefits: (i) the embedded components are parameterizable—allowing end-users to control the embedded energy and motion characteristics such as movement speed [32, 108] based on their design needs; (ii) kinematic mechanisms are auto-generated—end users do not require advanced mechanical design knowledge; and (iii) all embedded parts are 3D printed in place—reducing the need for manual assembly [32, 44].

Next, I describe an open-sourced interactive design tool—*Kinergy*—that allows end-users to embed the kinetic units into custom 3D models for desired output motions. With *Kinergy*, the user selects a target motion from seven supported motion types, customizes the motion characteristics (*e.g.*, motion direction, energy, movement) through a set of graphical user controls, adds a lock mechanism for motion actuation control, previews the action of the 3D model, and prepares the converted model for 3D printing on commodity 3D printers. For example, Figure 5.1 shows a pull-back car model that uses a *continuous rotation* kinetic unit for self-propelled motion. Next, we demonstrate the potential of our approach with a series of kinematic applications created with *Kinergy*, such as a self-opening umbrella

³<https://www.thingiverse.com/thing:38159>

(Figure 5.11), a custom pull-back car (Figure 5.14), and a self-lifting trash can (Figure 5.13). Finally, I validate *Kinergy* through a series of example applications created with the tool.

This chapter contributes (i) custom kinetic units for the parametric design and control of seven self-propelled motion behaviors with 3D printable springs, locks, and transmission mechanisms; (ii) an interactive design tool called *Kinergy* that lowers the barrier to creating highly custom motion-enabled models for 3D printing; and (iii) a variety of applications that show the potential of *Kinergy* to create motion-enabled devices.

5.1 Motion Types for 3D Printing

Traditional mechanical assemblies that interconnect kinematic elements include gears, cams, cranks, and levers—all of which support specialized movements such as translation, rotation, reciprocation, and oscillation. Amongst these motion types, *translation* describes motion along a fixed path, such as a sliding door; *rotational* motion is circular movement around an axis like a wheel on an axle; *reciprocating* motion describes an object repeatedly moves back and forth like a piston, and *oscillation* combines rotational and reciprocating motion like a pendulum.

In the 3D printing literature, past work has explored creating kinematic objects with 3D-printable motion behaviors, including translation [32, 108], rotation [121, 138], reciprocation [117, 56], and oscillation [108, 145]. However, these approaches support only one or a few motion types. In contrast, my approach explores three non-periodic 3D printable motions by producing a one-time, ephemeral movement following a line (instant and continuous translation) and an arc path (instant rotation). Additionally, *Kinergy* supports four repeated motion types by making the 3D-printed object rotate around an axis (continuous rotation), move back and forth following a linear trajectory (reciprocation), and move along an arc repeatedly (intermittent rotation and oscillation).

5.2 Kinetic Units

Kinetic units contain seven controllable output motions that are 3D-printable: *instant translation*, *instant rotation*, *continuous translation*, *continuous rotation*, *reciprocation*, *intermit-*

tent oscillation, and intermittent rotation. Each kinetic unit consists of an embedded energy source—either a helical or spiral spring, a compliant lock mechanism, and a transmission mechanism. Figure 5.2 shows all the kinetic unit compositions (rendered 3D models and cutaways) for target output motion types.

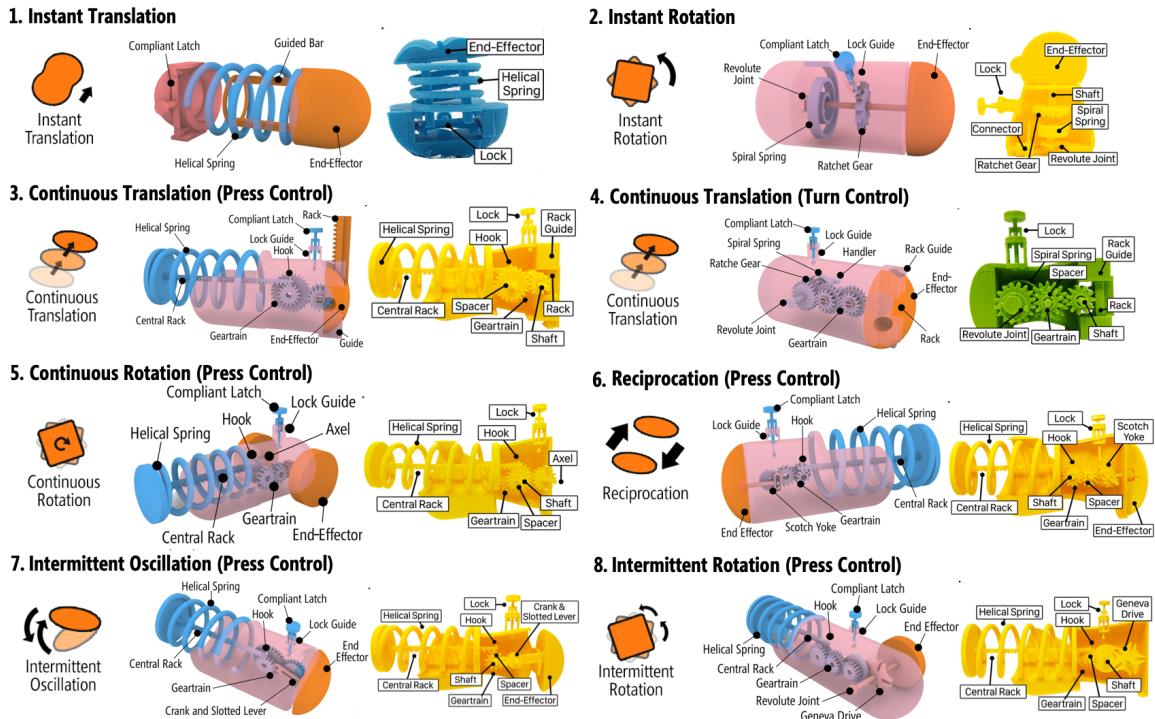


Figure 5.2: Motion types and *Kinergy* kinetic unit examples.

5.2.1 Spring Energy Sources

To enable self-propelled and controllable motion output, I use 3D printable and embeddable mechanical springs. As described in *Ondulé*, springs are attractive yet often overlooked energy sources for 3D printing. Inborn spring parameters such as coil diameter and the number of coils can be used to control the energy stored in deformed springs (Figure 2.1). In addition, the springs themselves can be 3D printed within an object, eliminating the need for other external actuators. As in prior work [108, 138], I use embedded helical and spiral springs, which can be customized based on spring parameters (Figure 2.1). To store the

potential energy in the spring, the user needs to either manually press on a helical spring or wind up a spiral spring. I demonstrate the capability of supporting these two energy-charging methods in two example applications: a pull-back car is activated by a winding spiral spring (Figure 5.14), and a handheld flashlight lights up by hand pressing (Figure 5.15).

5.2.2 Lock Mechanisms

To store the potential energy in the spring, kinetic units include unique in-place compliant structures are used as locks (Figure 5.3). The locks differ depending on the spring type.

For a *helical spring* energy source, there are two different designs in kinetic units. For the kinetic unit that supports the instant motion, two protruded guiding bars with notches are attached to the moving end of the spring, which snaps to two latch hooks situated in the stationary part when the spring is compressed (Figure 5.3a). The latch hooks are controlled via a compliant two-bar mechanism—pressing the button makes the two hooks move apart, releasing the two bars and unleashing the stored spring energy. For the kinetic units that enable non-instant motion, the helical spring has a central rack that acts as a moving hook and contacts and locks with a sliding latch when the spring is at its maximum compression (Figure 5.3b).

For a *spiral spring*, the spring can be locked at a rotation angle with a ratchet mechanism, which consists of one gear with asymmetrical teeth in parallel with the spring. A sliding compliant latch—working as the “pawl”—is mounted on the stationary part (Figure 5.3c). The latch uses an identical compliant two-bar design—two-sided hooks retract and move under button pressing and pulling and stop the sliding lock at positions when engaging grooves in the guided wall. Spring rotation is prevented by engaging one gear tooth with the inserted latch. The spiral spring turns and releases the stored potential energy by pulling the latch and freeing the ratchet gear.

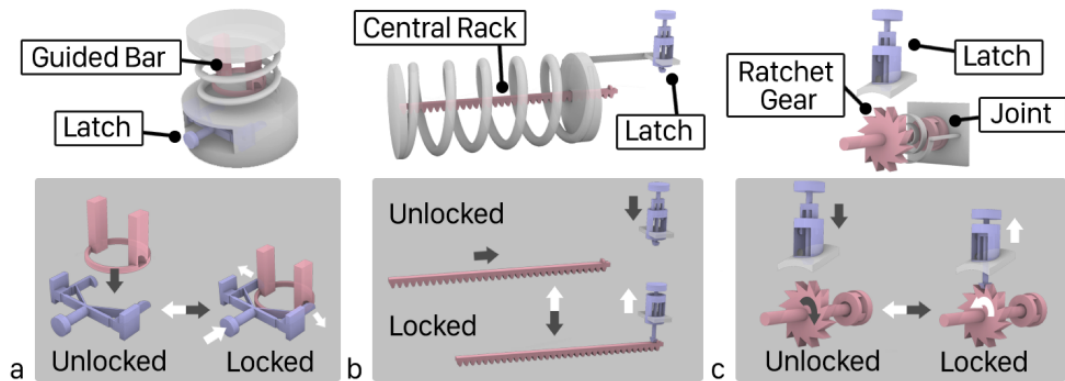


Figure 5.3: Compliant lock mechanisms used in kinetic units: (a) guided bars and compliant latch designs for the locking control of the helical spring and (b) ratchet gear and compliant latch design for the locking control of the spiral spring.

5.2.3 Transmission Mechanisms

To transform the driving force created from the spring into an output motion, we combine the spring energy source with specialized transmission mechanisms made of kinematic pairs. These pairs are joints between two contacting rigid mechanical components under the relative motion [2], including geartrain, rack-and-pinion, Scotch yoke, crank-and-slotted lever, and Geneva drive (Figure 5.4a-e). The specific mechanism is based on the desired movement enabled and the spring type used by the kinetic unit. For those non-instant motion types, a geartrain constructed with a series of two paralleled gears (a bigger bull gear and a smaller spur gear) is the crucial component to transmit the motion from the spring energy source to the end-effector.

To engage the spring with the geartrain, I use two approaches. For a helical spring, I attach a central rack to the helical spring and engage the rack with the first gear in the geartrain, which commits a rack-and-pinion mechanism (Figure 5.4f and Figure 5.2-helical spring-based *continuous translation* kinetic unit). For a spiral spring, the first gear in the geartrain and the spiral spring are coaxial, and the gear rotates when the spring is turned (Figure 5.4g and 5.2-spiral spring-based *continuous translation* kinetic units).

By combining the geartrain with different kinematic mechanisms, the end-effector can

achieve desired output motions (Figure 5.4a-e and 5.2): a rack that connects to the end-effector engages with the final gear in the geartrain to commit a *rack-and-pinion* for the output translation; a *revolute joint* or an axel rotates together with the final gear in the geartrain to drive the connected end-effector to rotate; the end-effector equipped with a *Scotch yoke* moves back and forth on a linear path; a *crank and slotted lever* design can transfer the rotary movement of the last gear in the geartrain into repeatedly oscillating movement, where the end-effector moves along an arc path; and finally, inserting a *Geneva drive* between the final gear in the geartrain and the end-effector leads to intermittent rotary movements of the end-effector.

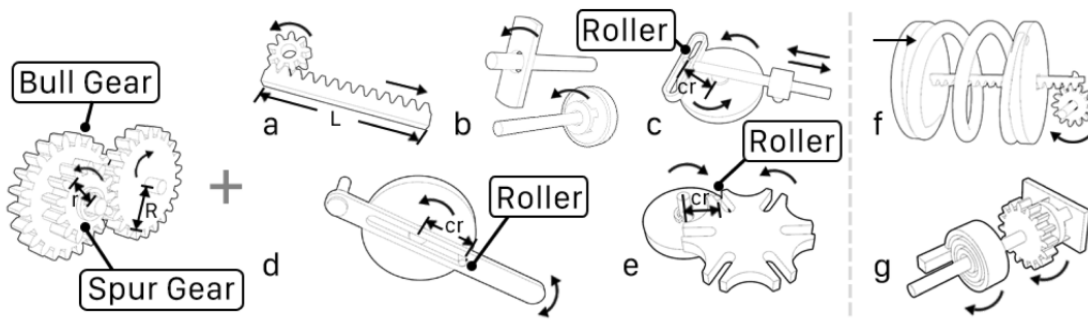


Figure 5.4: For non-instant motion types, kinetic units combine geartrains (pairs of bull and spur gears) and kinematic transmission mechanisms: (a) rack-and-pinions, (b) axels or revolute joints, (c) Scotch yokes, (d) crank-and-slotted-levers, and (e) Geneva drives. Helical springs engage with gears through (f) rack-and-pinions and (g) spiral springs are co-axial with gears for driving the geartrain.

5.2.4 Unit Types

The energy sources, locks, and transmission mechanisms described above are combined to create seven kinetic units (Figure 5.2). Each kinetic unit allows customization by parameterizing both the spring energy source and the transmission mechanism. Below, we enumerate the kinetic unit compositions and how they support distinct motion types.

Instant Translation. To enable an object to extend its body, the kinetic unit auto-replaces a portion of a selected 3D shape with a helical spring. The spring connects to

an end-effector and a stationary body segment (Figure 5.2-1). The converted spring is controlled by a compliant lock shown in Figure 5.3a. For example, a self-popping Halloween pumpkin décor has one instant translation kinetic unit embedded, and the sectional view is shown in Figure 5.2-1. Because the body directly executes the motion with a spring, no transmission mechanism is needed in this kinetic unit.

Instant Rotation. To enable an object to rotate, the kinetic unit auto-embeds a spiral spring in a selected object body, separating into two parts: an end-effector and a stationary segment (Figure 5.2-2). The spring center connects to the end-effector via a central shaft and the spring coil end connects to the stationary segment of the body via a solid rod. The central shaft extends toward the stationary part and connects via a *revolute joint* consisting of a bearing socket and a circular disc, avoiding extraneous shaft movement caused by turning. The converted spiral spring is controlled by a compliant lock shown in Figure 5.3c. Similar to *instant translation*, no transmission mechanism is needed.

Continuous Translation. To prolong the output translation motion, the kinetic unit uses a geartrain that joins with the spring energy source for motion transmission. A rack that connects to the end-effector mated with the last gear in the geartrain is used to perform as a rack-and-pinion for the output translation (Figure 5.2-3&4). This kinetic unit works with both helical and spiral spring energy sources. For helical spring control, the compliant lock uses the central rack as the guided bar rather than the original two-bar design in Figure 5.3b. For the spiral spring control, the first gear shaft extends and drills through the 3D body to provide a handler for spiral winding. The other end of the shaft is connected to the body via a revolute joint so that the shaft rotates in place. The placement of the spring energy source and lock mechanism are reused in other non-instant kinetic units.

Continuous Rotation. To support a continuous rotation output, the last gear in the geartrain and the end-effector are coaxial on a shaft, which drives the end-effector to rotate. If multiple end-effectors reside on the opposite sides of the 3D body, the last gear shaft extends and drills through the body as an axel to connect end-effectors; otherwise, the shaft only extends in one direction, and the other end connects to the body via a *revolute joint* (Figure 5.2-5).

Reciprocation. To create a reciprocating motion, a Scotch yoke (*a.k.a.*, slotted link

mechanism) is used to connect the last gear in the geartrain and the end-effector (Figure 5.2-6). The Scotch yoke comprises a circular disk, a roller, a yoke, and a connecting rod. The circular disk resides coaxially with the final gear and the connecting rod connects to the end-effector. The linear motion guides are fixed to the 3D body. When the circular disk rotates with the gear, the roller slides inside the yoke, making the connecting rod and end-effector move repeatedly.

Intermittent Oscillation. To enable an intermittent oscillation, a crank-and-slotted lever (one type of quick-return mechanism) is driven by the final gear in the geartrain to actuate the end-effector to oscillate along an arc path (Figure 5.2-7). The crank-and-slotted lever consists of a pivot that is fixed inside the 3D body, a bull gear that resides coaxially with the final gear in the geartrain, a crank pin, and a slotted bar that connects to the end-effector. When the bull gear turns, the crank pin slides back and forth in the slotted bar and swings the end-effector around the pivot for intermittent oscillating motion.

Intermittent Rotation. To create an intermittent rotation, a Geneva drive, which translates a continuous rotational motion into an intermittent rotational motion, resides coaxially with the final gear in the gear train (Figure 5.2-8). The Geneva drive consists of two parts: a driving and driven wheels. When the driving wheel rotates, the protruded roller on the driving wheel goes in and out of the slot on the driven wheel repeatedly, resulting in an intermittent rotational motion. The end-effector also rotates intermittently because it shares the same shaft with the driven wheel.

5.3 *Kinergy Design Tool*

Kinetic units are the foundations to lower design barriers for complex mechanical motions. However, to integrate these motions into 3D models, kinetic units need to be customizable with a front-end interactive design tool, *Kinergy*. *Kinergy* is an open-source plugin for Rhino 6 (Figure 5.5), with the front-end user interface built with *Grasshopper*⁴ and *Human UI*⁵, and the backend written in C# using the RhinoCommon API.

⁴Grasshopper: <https://www.rhino3d.com/6/new/grasshopper>

⁵Human UI: <https://www.food4rhino.com/app/human-ui>

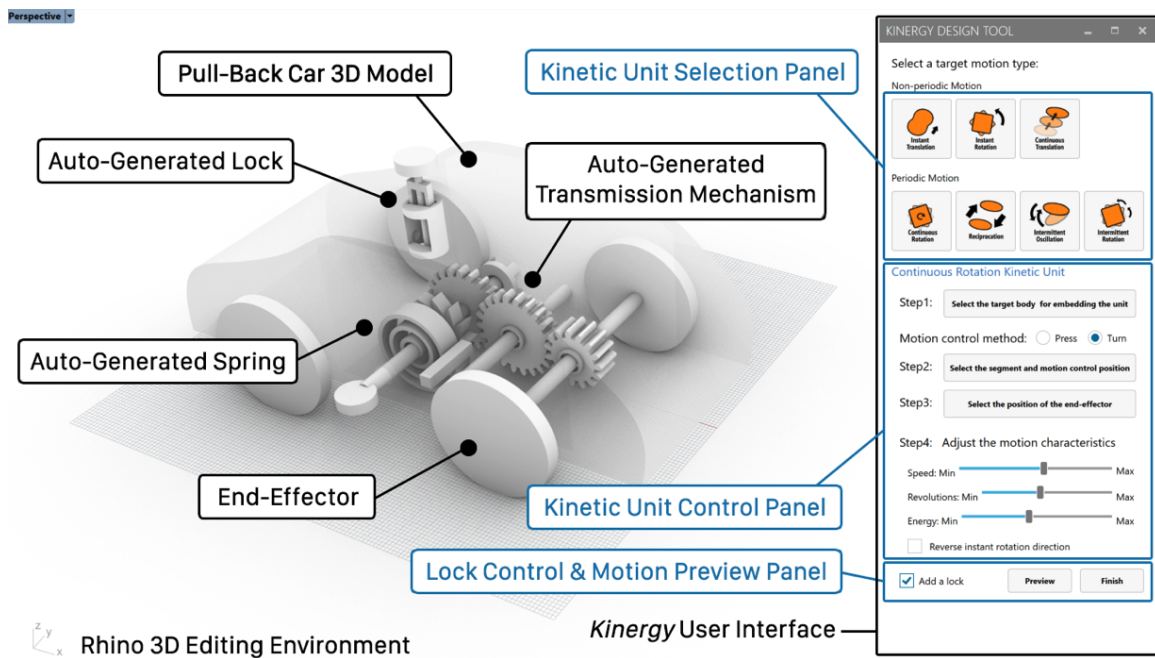


Figure 5.5: The user interface of *Kinery* design tool.

Kinery consists of three parts (Figure 5.5): a *Kinetic Unit Selection* panel, *Kinetic Unit Control* panel, and *Lock Control and Motion Preview* panel. The *Kinetic Unit Selection* panel (Figure 5.5) provides seven buttons, each indicating the supported motion type and kinetic unit. The *Kinetic Control* panel (Figure 5.5) displays a series of steps to complete the embedding of the selected kinetic unit and to parameterize motion behaviors, such as energy strength and motion displacement adjustment. Finally, the *Lock Control and Motion Preview* (Figure 5.5) panel allow the user to add a lock to the spring for motion control and preview the generated motion behavior via a simulated animation in a separate window. Below, I describe the workflow of the design tool through a pull-back car, which uses a continuous rotation kinetic unit. I also describe three key parts of the tool: the auto-generation of kinetic units, the parameterization of kinetic unit mechanisms, and motion previews.

5.3.1 Design Walkthrough: Creating a 3D Printable Pull-Back Car

To create a pull-back car model that uses spring energy to self-propel, the user needs to embed a continuous rotation kinetic unit into a 3D car model. See our video, which complements the following description.

1. In the Rhino 3D editing environment, the user first creates a 3D pull-back car model that includes a car body and four wheels (Figure 5.6a). The four wheels are positioned in parallel with the car body. The user aligns the facing direction of the car body with the X-axis in the 3D environment.
2. The user clicks on the *Continuous Rotation Kinetic Unit* button from the *Kinetic Unit Selection* panel and Kinetic Unit Control panel, then displays the instructional user controls (see the buttons and sliders in Figure 5.6a).
3. Then, the user selects the target car body in the Rhino 3D editing environment and confirms the selection by clicking on *Select the target body for embedding the unit* button (Figure 5.6b).
4. The user selects the motion control method—turn (supported by a spiral spring energy source).
5. After the control method is confirmed, the user clicks on the *Select the segment and motion control position* button, and three colored axes appear for the user to select the car's body orientation, which is the X-axis (Figure 6b). Upon selecting the body orientation, two adjustable paralleled planes perpendicular to the body orientation appear. The user can select the target body region for embedding the kinetic unit by dragging these two planes separately, and the planes move along the X-axis (Figure 5.6b). Once the portion is confirmed, the 3D body is split into three parts: two end parts and the middle body that contains the embedded kinetic unit. The user needs to select one end part to indicate which side the spring of the kinetic unit will be placed on (Figure 5.6b).

6. Then, the user selects the two wheels next to the other end part (not the spring side) as the end-effectors after clicking on the *Select the position of the end-effector* button (Figure 5.6c). Upon the selection of the end-effector, the continuous rotation kinetic unit, including the spring energy source, the geartrain, and the revolute joint, is generated automatically inside the 3D body.
7. After the kinetic unit is generated, the user can adjust the needed energy, speed, and rotating revolutions by dragging the sliders on the user interface (Figure 5.6d). The spring design changes are based on the energy adjustments and the geartrain updates with user changes on the speed and revolution sliders in real-time in the 3D editing environment.
8. Optionally, the user can add a spring lock by checking the *Add a lock* checkbox on the interface (Figure 5.6e). Once the checkbox is checked, a ratchet gear and the compliant lock mechanism are auto-generated in place.
9. Finally, the user examines how the pull-back model moves and the wheels rotate through an animation of the motion in a separately popped window after clicking on the *Preview* button (Figure 6f). In this window, the user views the converted pull-back car model in interactive ways: rotating, panning, and zooming. To trigger the motion preview, the user first clicks on the *Load Motion* button to charge the energy in the spring. Then, the user releases the energy by clicking on the *Release* button and the simulated animation begins. All the generated 3D parts can be exported for 3D printing by hitting *the Finish* button (Figure 5.6g).

5.3.2 Generating Kinetic Units

Generating kinetic units contains three processing steps: determine the position and orientation of the embedded kinetic unit, generate the kinetic unit components (spring energy source, transmission mechanism, and lock), and interface the kinetic unit with the 3D body.

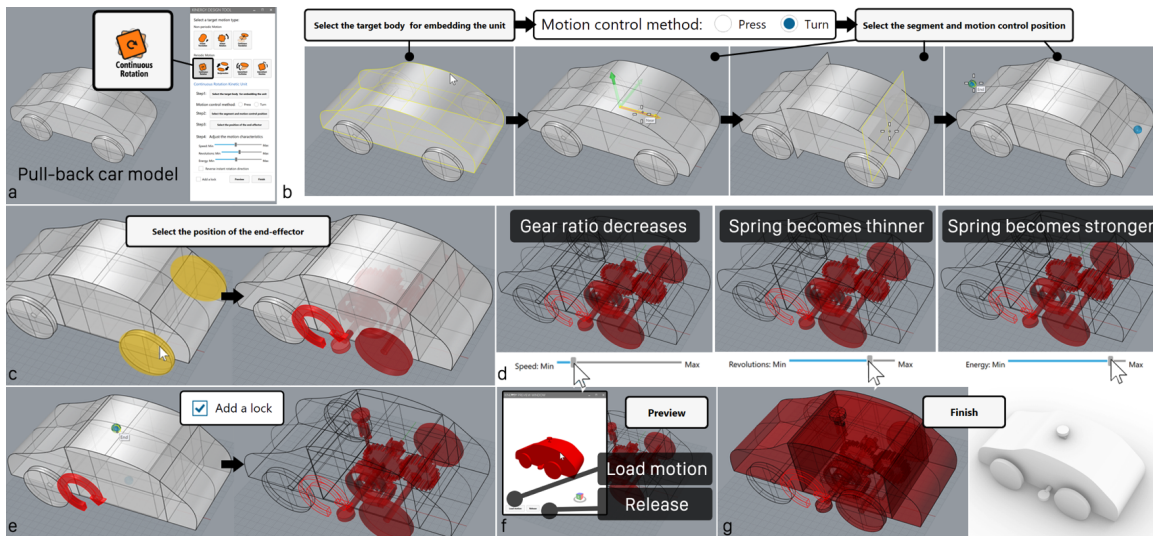


Figure 5.6: The unit orientation, translation axes and rotation axes in each kinetic unit, and end-effectors and stationary parts.

Below, I describe these steps by starting with a set of terms that we use in our implementation: Dir_p denotes the pose direction of the embedded kinetic unit, $Axis_{trans}$ denotes the axis that a part translates along, $Axis_{rot}$ denotes the axis that a part rotates around, $Part_{st}$ denotes the stationary part that connects to the kinetic unit, and $Part_{ee}$ denotes the end-effector that connects to the kinetic unit (Figure 5.7).

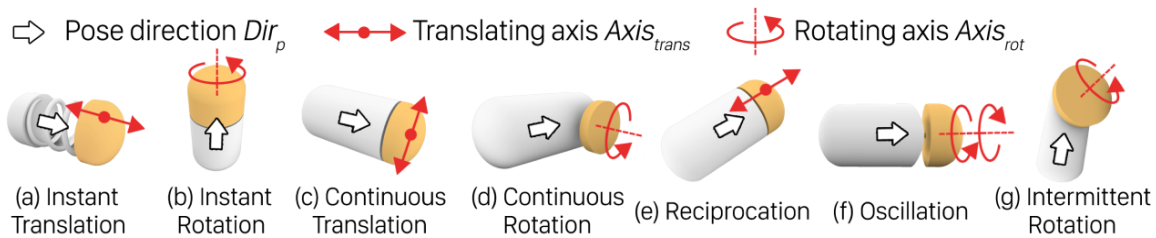


Figure 5.7: The unit orientation, translation axis and rotation axis in each kinetic unit. The end-effectors are highlighted in light orange and the stationary parts are marked in grey.

The tool first decides the position and orientation of the embedded kinetic unit based on the user selection of the $Part_{ee}$ and user input directions. All the kinetic units are embedded

in a user-selected segment of the 3D body (see Design Walkthrough step 4). For kinetic units with no geartrain, the selected body portion is converted with an embedded spring,. The Dir_p of the kinetic unit is aligned with either the helical spring's translation axis or the spiral spring's rotation axis (Figure 5.7a-b). Once the Dir_p is determined, the kinetic unit is generated in place regardless of which direction the unit is oriented in. For kinetic units that use geartrain and transmission mechanisms, Dir_p is aligned with the propagating direction of the gears in the geartrain (Figure 5.7c-g). For those non-instant kinetic units, the unit orientation also depends on the rotary angle along with the Dir_p , which is input by the user in the tool.

After the position and orientation of the embedded kinetic unit are determined, *Kinergy* computes and generates the spring, transmission mechanism, and lock. For instant kinetic units, the spring resides close to the center of the selected segment. Then, by computing the user input energy and motion attributes (*e.g.*, translation displacement, rotation angle), the tool uses the *RhinoCommon spiral* function to create a spiral curve and the *sweep* function to create a solid spring. Upon selecting the lock positions by the user, *Kinergy* auto-generates the lock structures in place.

For those non-instant kinetic units, *Kinergy* also generates transmission mechanisms in addition to springs and locks. To create the geartrain, the tool first determines the number of gear sets and gear positions based on the body segment volume and Dir_p . All the bull and spur gears in the geartrain share the same gear module, which is the unit of size that indicates how big and small gear is, in our implementation. Then, *Kinergy* generates all the gear models using gear parameters such as gear diameter and the number of teeth, which are determined by the input motion speed, and the approach of calculating an involute spur gear⁶. All the gears automatically self-rotate to engage with each other. While the tool generates a spiral spring that co-axially resides with the input gear (the first gear) in the geartrain for a *Turn* control, a helical spring is generated and mated with the input gear of the geartrain via an auto-generated teathed rack for a *Press* control. Lastly, to generate the specialized kinematic elements for specific motion purposes, *the* design tool generates the

⁶Calculation of involute gears: <https://www.tec-science.com/mechanical-power-transmission/involute-gear/calculation-of-involute-gears/>

parametric 3D models of those specialized elements in place to engage with the output gear (the last gear) in the geartrain. The $Part_{ee}$ is connected to a kinematic element that moves by strictly following either a translation axis or a rotary axis. For example, the $Part_{ee}$ is attached to a toothed rack that mates with the output gear and travels along a linear path in a continuous translation kinetic unit. The $Part_{ee}$ resides on the central shaft for the driven wheel in an intermittent rotation kinetic unit. Hence, the $Part_{ee}$ rotates repeatedly along with the Geneva drive.

Finally, the design tool also generates additional structures to secure the kinetic unit in the 3D model and to ease loading the energy in the springs. For example, the gear shafts in the geartrain are auto-extended to connect both ends to the solid $Part_{st}$. A solid pole is auto-generated to fixate the outer end of the spiral spring to the $Part_{st}$ in the kinetic units. A pair of spacers are auto-generated on both sides of the gear to prevent the movable gears from sliding on the shafts. For non-instant kinetic units, the tool creates a thin cylinder as the button to compress the helical spring and graspable knobs as the key to wind the spiral spring.

5.3.3 *Parameterizing Embedded Energy and Motion Properties*

Kinergy design tool allows the user to control stored energy in 3D models by adjusting the spring parameters and characterize desired motions by changing the parameters in the geartrain and the specialized kinematic elements. Table 5.1 shows all the parameters used for energy control and motion characterization in the design tool.

Informed by spring theory (p. 156 and p. 537 in [11]), the potential energy stored in the deformed springs are impacted by spring parameters and the amount of spring deformation. For the helical spring, the potential energy that the spring can achieve is proportional to the fourth power of coil thickness d and the square of the compression/extension displacement X , while inversely proportional to the third power of spring diameter D and the number of coils N (Eq. 5.1). For the spiral spring, the potential energy that the spiral spring can achieve is proportional to the spring width b , the third power of the thickness t , and the square of rotary angle θ , while inversely proportional to the number of coils N (Eq. 5.2).

Table 5.1: Parameters for energy control and motion characterization in the kinetic units.

Kinetic Unit Type	Adjustable Parameters	Controllable Output	Relationship
Instant translation	Spring diameter D , coil thickness d , the number of coils N , compression/extension displacement X	Helical spring energy P_e , spring displacement X	$P_e \propto \frac{d^4 X^2}{D^3 N}$ (5.1)
Instant rotation	Spring width b , thickness t , rotary angle θ , the number of coils N	Spiral spring energy P_e , rotating revolutions Rev	$P_e \propto \frac{bt^3 \theta^2}{N}$ (5.2) $Rev \propto \theta$ (5.3)
Continuous translation	Gear ratio of the geartrain G_{ratio} (bull gear radius R , spur gear radius r , number of bull and spur gear sets N), spring parameters (see above)	Spring energy P_e , motion speed S , translating distance Dis	$S \propto G_{ratio} = \left(\frac{R}{r}\right)^N$ (5.4) $Dis \propto \begin{cases} X \times G_{ratio} \\ \theta \times G_{ratio} \end{cases}$ (5.5)
Continuous rotation	Gear ratio of the geartrain G_{ratio} , spring parameters	Spring energy P_e , motion speed S , rotating revolution R_r	$R_r \propto \begin{cases} X \times G_{ratio} \\ \theta \times G_{ratio} \end{cases}$ (5.6)
Reciprocation	Gear ratio of the geartrain G_{ratio} , spring parameters, Scotch yoke crank length r	Spring energy P_e , motion speed S , stroke Str , reciprocating distance Dis_{rec}	$Str \propto \begin{cases} X \times G_{ratio} \\ \theta \times G_{ratio} \end{cases}$ (5.7) $Dis_{rec} \propto r$ (5.8)
Intermittent Oscillation	Gear ratio of the geartrain G_{ratio} , spring parameters, quick-return crank length r	Spring energy P_e , motion speed S , stroke Str , oscillating amplitude Amp	$Amp \propto r$ (5.9)
Intermittent Rotation	Gear ratio of the geartrain G_{ratio} , spring parameters, number of opening slots on the driven wheel n	Spring energy P_e , motion speed S , stroke Str , interval angle θ_{int}	$\theta_{int} \propto \frac{1}{n}$ (5.10)

These parameters inform the customization of energy applied in 3D models in *Kinergy*.

Besides energy control, the design tool allows the user to customize motion characteristics through sliders in the user interface (Figure 5.5). For instant translation kinetic unit, the user can specify spring compression by directly controlling the displacement values through a slider. Similarly, the user can edit the rotating revolutions of the spiral spring, which is proportional to the rotary angle θ (Eq. 5.3), for an instant rotation kinetic unit. For non-instant motions, the user can control the motion speed, which relates to the gear ratio of the geartrain. As the speed slider moves, the tool recalculates and updates the gears in the geartrain, including the bull gear radius R , spur gear radius r , and the number of bull and spur gear set N in the geartrain (Eq. 5.4). For example, when the speed increases, the tool updates the geartrain by creating more gear sets with a bigger radius discrepancy between the bull and spur gears. In addition to speed control, the design tool also supports specialized motion characterizations for those non-instant kinetic units via user-controllable sliders. For the continuous translation and rotation, the translating distance and rotation revolutions are proportional to the spring deformation amount and the gear ratio of the geartrain (Eq. 5.5&5.6). For reciprocation and oscillation, the reciprocating distance and oscillating amplitude are also proportional to the crank length in their Scotch yoke and crank-n-slotted-lever designs (Eq. 5.8&5.9). Finally, for intermittent rotation, the rotary angle at each rotation step is inversely proportional to the number of opening slots in the driven wheel of the Geneva drive (Eq. 5.10). For example, the driven wheel generates more slots as the user drags the interval angle slider toward the left side, indicating a smaller rotary angle at intervals.

5.3.4 Previewing Generated Motion

Kinergy design tool provides the motion preview by encoding all the part types and interactions between parts in a graph, which is similar to [75, 145]. In the graph, each node stores the part type (*e.g.*, spring, gear, rack-and-pinion, non-kinematic connectors), part parameters (*e.g.*, spring wire diameter, number of coils), and its motion attributes (*e.g.*, rotation axis). In addition, each edge encodes one of the three interaction types between

two mechanical parts: fixation (*e.g.*, a gear shaft is fixed to the solid 3D body), engagement (*e.g.*, two gears mate with each other), and locking (*e.g.*, a compliant latch stays in the lock groove). The kinetic unit created with our tool is represented using a graph; for example, Figure 5.8 shows a graph representation of a pull-back car with a continuous rotation kinetic unit embedded.

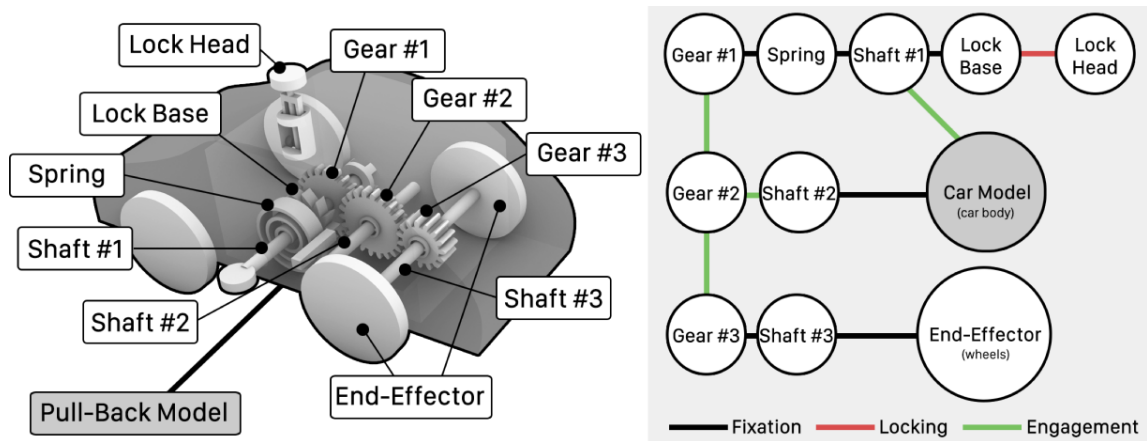


Figure 5.8: The pull-back car 3D model is represented by a graph for motion preview.

With the graph-based representations, the design tool creates an animated simulation for the motion under three assumptions: (i) the driving force solely originates from the spring deformations; (ii) external forces such as frictions and object weight are negligible in the simulated animation; and (iii) the springs deform realistically and strictly, *e.g.*, helical springs only compress/extend along a linear path and spiral springs only rotate around their central axes. The animation is rendered at a 20 FPS frame rate and displayed in a separate window. In each frame, the tool begins with the spring node by calculating the spring movement using *Hooke's Law*—capturing the amount of spring deformation, which derives motion acceleration and speed. Then, spring deformation changes are propagated to all the graph nodes through edges using depth-first searching (DFS). During the search, the part position and speed stored in each node are updated based on its part type, interaction with a neighbor node, and position changes from the neighbor. Finally, the transmissions terminate when the spring returns to its equilibrium or a new locking is applied, *i.e.*, the

user clicks the *Load Motion* button and the spring is locked again.

5.4 Fabrication

Kinergy aims to add self-propelled motion behaviors that can be 3D printed without post-print manual assembly. To achieve this goal, I provided calibrated tolerances for the printability of kinetic units and guidance for slicing. The calibration data is based on a dual-extruder Ultimaker 3D printer, which uses PLA as the printing material and PVA as the support material. First, I identified three types of mechanical gaps in a kinetic unit design that may be problematic for consumer-grade 3D printing (Figure 5.9a-c): the gap between two engaged gears (Type1), between the gear and its nearby spacer on the shaft (Type2), and between the gear and its residing shaft (Type3). To ensure that robust and functional gears and spacers are printed on the shafts in one shot, I took a trial-and-error approach and found that the printing orientation, as shown in Figure 5.9d, is superior to the other directions due to anisotropic 3D printing. For example, with a printer that uses a 0.4mm-sized printing nozzle and 0.15mm printing layer height, I found that a 0.3mm Type1 gap makes sufficient room for the gear backlash, which is the distance between the involutes of the mating gear teeth, and a 0.25mm Type2 gap and a 0.35mm Type3 gap appropriate as minimal distances. Further, the 3D model is sliced with an Exclusive slicing tolerance and a minimal negative 0.05mm horizontal expansion in a slicer software (*e.g.*, Cura) to avoid over-fused gaps and create a clean profile contour for the inter-engaged parts such as gears (Figure 5.9d).

To examine if our approach applies to other 3D printers and printing methods, I also printed pull-back car models created with *Kinergy* on 3D printers that offer different printing capabilities (Figure 5.10): an industrial FDM 3D printer (Stratasys F170; printing material: ABS; support material: water-soluble QSR), an industrial PolyJet 3D printer (Stratasys J750; printing material: ABS; support material: SUP705), and a desktop SLA-based 3D printer (Form 3; printing and support material: Resin). As a result, the printed cars are functional except for the one printed with the single material-based approach (Figure 5.10c), which has support residuals in the car body and with some imperfect printing finishes. I also found that printing tolerance varies across all these methods. For example, the same set

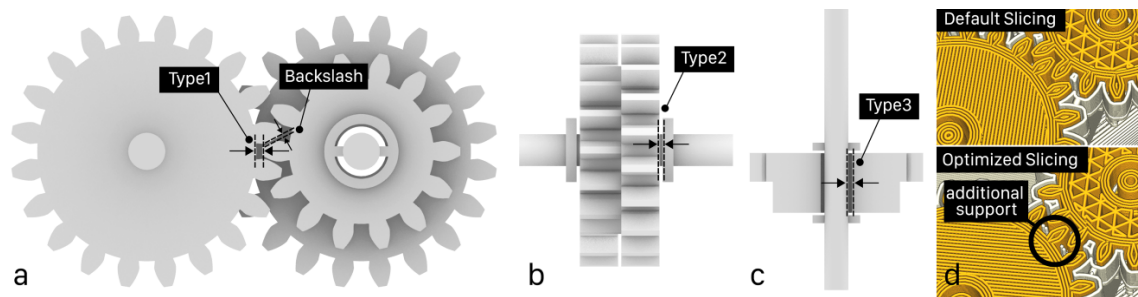


Figure 5.9: Three types of tolerance identified as problematic for one-shot printing: (a) the gap between two mating gears, (b) the gap between a gear and its nearby spacer, and (c) the gap between a gear and its residing shaft. The 3D model is sliced in (d) an optimized orientation and the slicing settings are curated to create clean part contours for intermating elements.

of Type1-3 tolerances works perfectly on the PolyJet-based print (Figure 5.10b), while the gears and joints are wiggly in the car printed with the industrial FDM 3D printer (Figure 5.10a). Based on these preliminary explorations, I anecdotally conclude that our approach is feasible with various multi-material 3D printing technologies with appropriate printing tolerances.

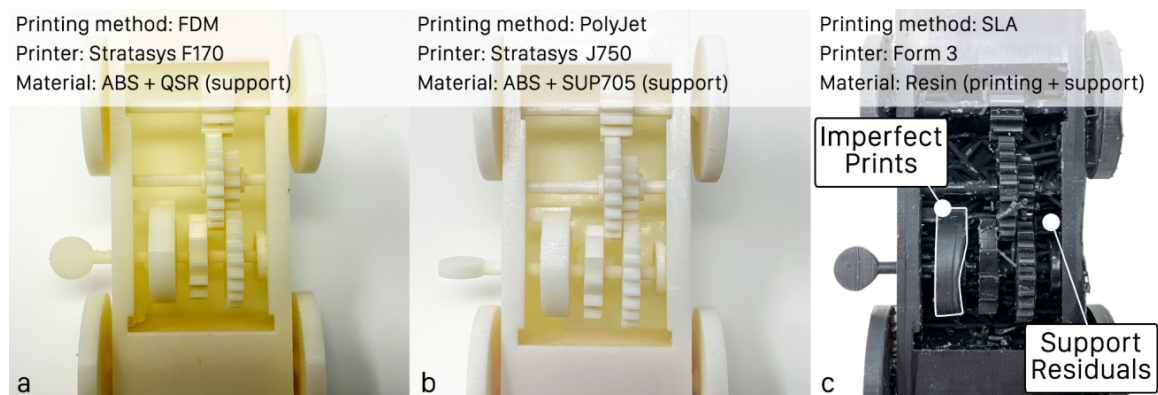


Figure 5.10: The pull-back cars created with *Kinergy* are printed with various 3D printers and printing technologies: (a) industrial-level FDM 3D printing, (b) PolyJet 3D printing, and (c) SLA 3D printing.

5.5 Applications

To demonstrate the potential of *Kinergy*, I created eight functional models with *Kinergy* and showcased how each kinetic unit is used to support a specialized motion in these applications: an *instant translation* kinetic unit is used in a self-opening umbrella, an *instant rotation* kinetic unit is used in a game controller, a *continuous translation* kinetic unit is used in a trash bin, a *continuous rotation* kinetic unit controlled by a turning input is used in a pull-back car, a *continuous rotation* kinetic unit is used in a handheld flashlight, a *reciprocation* kinetic unit is used in a lucky cat model, an *oscillation* kinetic unit is used in a cutter, and an *intermittent rotation* kinetic unit is used in a zoetrope sculpture. All the models are printed in one shot, and the dissolvable PVA support is entirely removed before the demonstration.

5.5.1 Auto-Opening Umbrella

A spring-loaded umbrella or parasol includes a folding canopy supported by jointed ribs mounted to a pole, which automatically opens at the press of a button. To create an auto-opening umbrella prototype with *Kinergy*, I can use the *instant translation* kinetic unit. The prototype consists of a spring-loaded runner and a slider that moves along the pole and connects to the jointed ribs (Figure 5.11a). The runner and the slider with jointed ribs were printed separately, and the runner was mounted on the pole beneath the slider. With *Kinergy*, I created the runner by converting a cylinder into a compressible helical spring with an embedded lock. The spring is locked at its maximum compression and the slider stays on top of the spring when the umbrella is closed (Figure 5.11b). The user presses the built-in lock button to open the umbrella, and immediately, the spring in the runner extends, pushing the slider forward and unfolding the umbrella ribs (Figure 5.11c). A canopy cloth could be added but is not included to avoid occlusion.

5.5.2 Angle Adjustable Game Controller

I created a catapult-like launcher for an Angry Birds game to demonstrate the *instant rotation* kinetic unit and the ability to use *Kinergy* to design custom game controllers

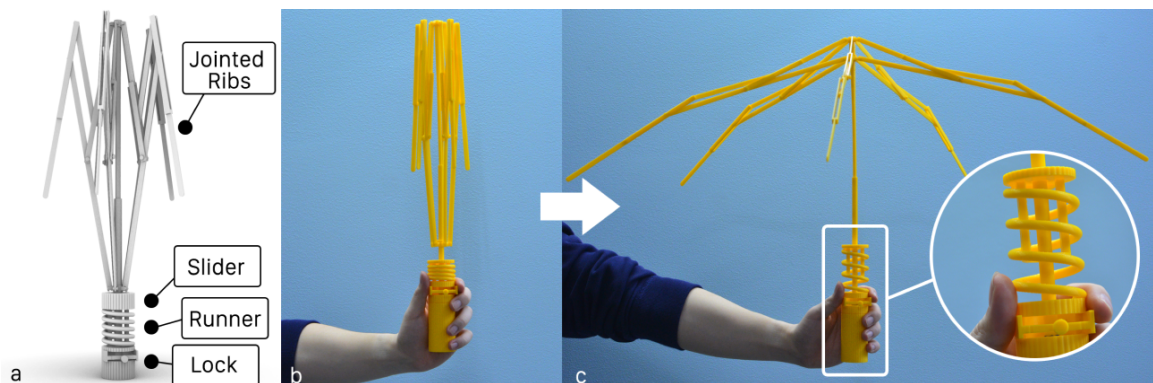


Figure 5.11: An auto-opening umbrella prototype created with the instant translation kinetic unit: (a) the rendered 3D model of the umbrella, (b) the printed and assembled umbrella in the locked stat, and (c) the opened umbrella after the runner is unlocked.

quickly. First, I embedded a spiral spring in the launcher body to create a stationary base and a twistable shooting arm (Figure 5.12a). The arm can be rotated and locked at different positions for desired shooting angle. Next, I mounted an accelerometer on the controller's arm to calculate the shooting angle, which was used to simulate the bird flying trajectory in a game built with Processing (Figure 5.12b-d). To play the game, the player first turns the shooter's arm to a certain angle based on how far the shooter is from the target and then locks the arm at the angle, waiting for a bird to fly into the launching area (Figure 5.12d). When a bird is in the launching area, the player releases the locked arm and projectiles the virtual bird toward the target.



Figure 5.12: A catapult-like game controller is made to virtually projectile birds in an Angry Birds game: (a) the rendered 3D model of the game controller, (b) the printed game controller with external sensor and circuitry embedded, (c) flying the bird at a small angle, and (d) flying the bird at a bigger angle.

5.5.3 Self-Actuated Trash Can

For the last non-periodic motion—continuous translation, I built a self-actuated trash can by attaching a *continuous translation* kinetic unit-embedded switch on one side of a cardboard trash can (Figure 5.13). The switch has a helical spring-based button and the rear end of the switch is connected to the lid of the trash can (Figure 5.13a-b). To close the can, the user presses the switch button and locks the switch by inserting the built-in latch (Figure 5.13c). When the user pulls the latch, the compressed switch button extends immediately to drive the link arm to translate upward, lifting the can’s lid and opening the trash can (Figure 5.13d).

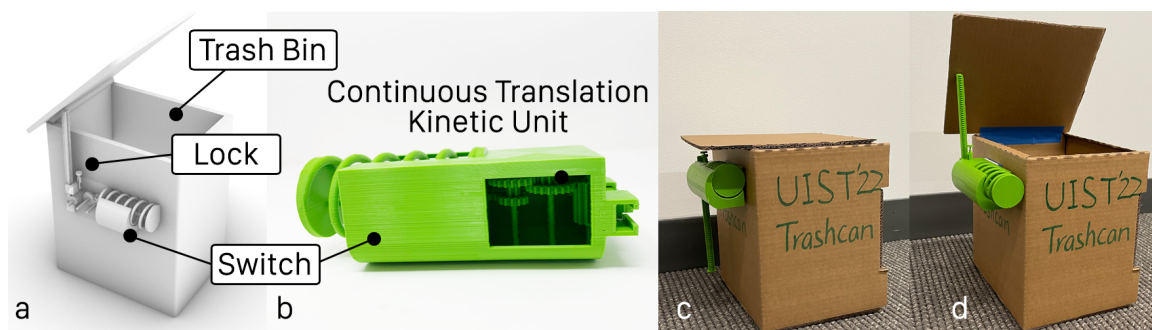


Figure 5.13: A continuous translation kinetic unit embedded switch is attached to a cardboard trash bin: (a) the rendered 3D model of the switch and the trash bin, (b) the printed switch, (c) the closed trash bin with the switch attached and locked, and (d) opening the bin’s lid by unlocking the switch.

5.5.4 Motion Parameterizable Pull-Back Cars

To demonstrate how to control energy and motion with *Kinergy*, I created four pull-back cars by varying the embedded energy, motion speed, and travel distance in separate continuous rotation kinetic units (Figure 5.14). First, I added a *continuous rotation* kinetic unit with low energy, low speed, and fewer achievable rotary revolutions to the 3D car model as a baseline (see the 3D model and the printed car in yellow in Figure 5.14a). Then, to show how *Kinergy* allows the user to achieve motion with various embedded energy, I compare how far the baseline model can hit a paper cup with a printed car in green, which has

double energy embedded in the kinetic units through the design tool (Figure 5.14b). As a result, the comparison model hits the paper cup double farther than the baseline model. For the speed comparison, I built another car (in blue) that travels faster than the baseline model (Figure 5.14c). Finally, I also compared the baseline model with another car model (in orange) that had a kinetic unit embedded for achieving more than three times rotary revolutions (Figure 5.14d). As predicted, the orange car travels a longer distance than the baseline model. These pull-back car examples validate that *Kinergy* allows users to parameterize embedded energy and motion characteristics for 3D printing.

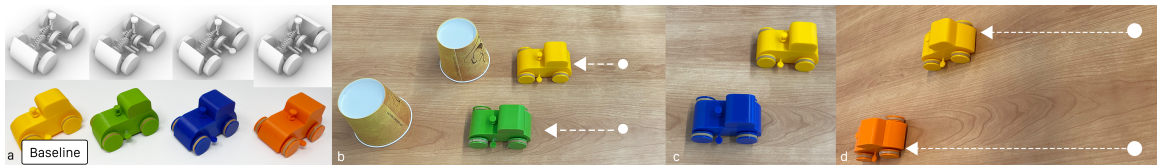


Figure 5.14: Four 3D-printed pull-back cars with different embedded energy, speed, and traveling distance: (a) the rendered car models and the corresponding printed cars (yellow: baseline; green: more stored energy; blue: faster; orange: longer travel distance), (b) the comparison of cars with different embedded energy, (c) the comparison of cars with different motion speed, and (d) the comparison of cars with different traveling distance.

5.5.5 Human-Operated Handheld Flashlight

Unlike the pull-back cars driven by winding springs, I created a human-operated handheld flashlight to demonstrate pressing as the energy charging method in the *continuous rotation* kinetic unit (Figure 5.15). I used a helical spring motor in this example that stores potential energy under compression. An electromotor wired to a LED is mounted and fixed in the flashlight head, and the motor's axle is inserted into a socket that is driven by the gears in the kinetic unit (Figure 5.15b). When the user presses the spring, the motor rotates its axle and generates current to light up the LED light (Figure 5.15c). Since the lock and release mechanism was not added to this device, the user can repeatedly produce lasting light by pressing the flashlight.

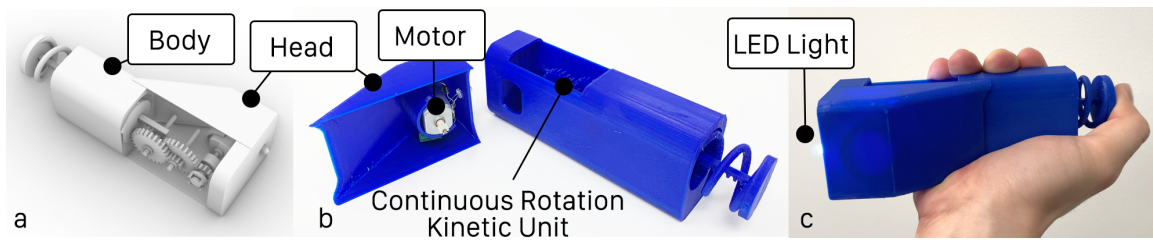


Figure 5.15: A human-operated handheld flashlight created with *Kinergy* and external electronics: (a) the rendered model of the handheld flashlight, (b) the flashlight head and body, and (c) the functioning flashlight with the embedded kinetic unit.

5.5.6 Battery-Free Maneki Neko Sculpture

To demonstrate the oscillating movement in a 3D printable device, I created a waving arm actuated by an embedded *oscillation* kinetic unit for a Maneki Neko sculpture (Figure 5.16). The embedded kinetic unit is designed with the maximum oscillation amplitude and the number of strokes to provide expressive arm movements. After the arm is inserted into the cat's body, the user winds the spiral spring from the side handler to load the energy. Then, without a lock, the arm begins to swing as the user releases the winded spring and stops swinging when the loaded energy dissipates.

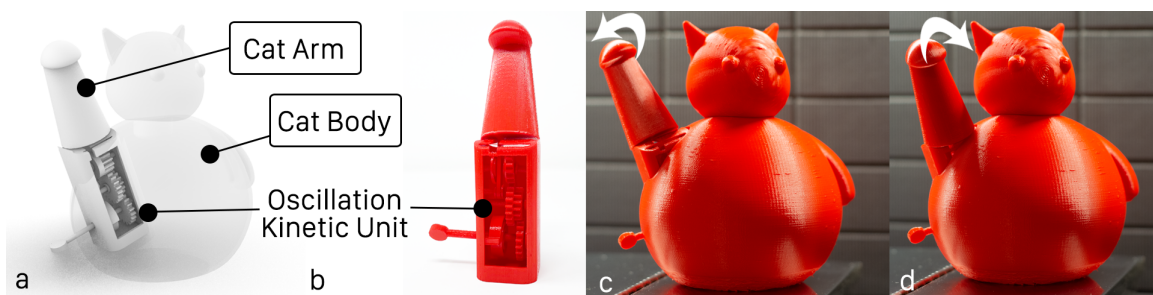


Figure 5.16: A battery-free Maneki Neko sculpture with an embedded oscillation kinetic unit: (a) the rendered model of the cat sculpture, (b) the printed sculpture, and (c) the waving arm driven by the embedded kinetic unit.

5.5.7 Semi-Automated Cutter

I created an assistive cutter that uses an embedded *reciprocation* kinetic unit to relieve people from the fatigue of the repeated back-and-forth cutting actions (Figure 17). The cutter embeds the kinetic unit into an organic, easy-to-hold shape and has a blade attached to the head (Figure 5.17a-b). The user first loads energy by turning the unit's spiral spring and then releases the lock at the top to execute the cutting job—the blade moves back and forth repetitively (Figure 5.17c-d). Like the other applications, since the sole power source is the embedded spring, the user needs to re-charge the energy in the spring for repetitive cutting tasks.

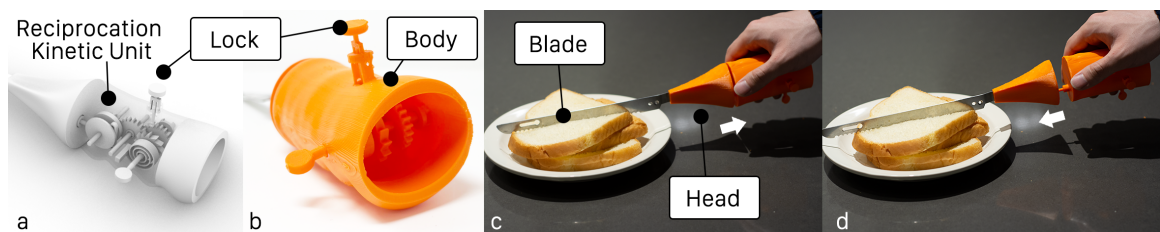


Figure 5.17: The user holds a 3D-printed cutter that uses an embedded reciprocation kinetic unit to move the blade back and forth repeatedly and cut the bread: (a) the rendered model of the cutter, (b) the printed cutter, and (c) the functioning cutter.

5.5.8 3D-Printed Zoetrope

Finally, I used an *intermittent rotation* kinetic unit in a box that connects and spins a circular scaffolding of six 3D-printed cat models to present a physical zoetrope installation—an animation of a walking cat (Figure 5.18). To create the physical animation, I first used a Geneva drive with six opening slots to drive the scaffolding that also has six cat models on the circumference (Figure 5.18a-b). I then mounted the zoetrope model on a helping hand soldering station and installed four bright LED lights around the model that were programmed through Arduino to provide interval light as the zoetrope spun (Figure 5.18c). Before the spinning, I loaded the energy in the base box by winding the spiral spring and started the LED control program. After the lock was released, the zoetrope model spun

and created the illusion of a walking cat (Figure 5.18d).

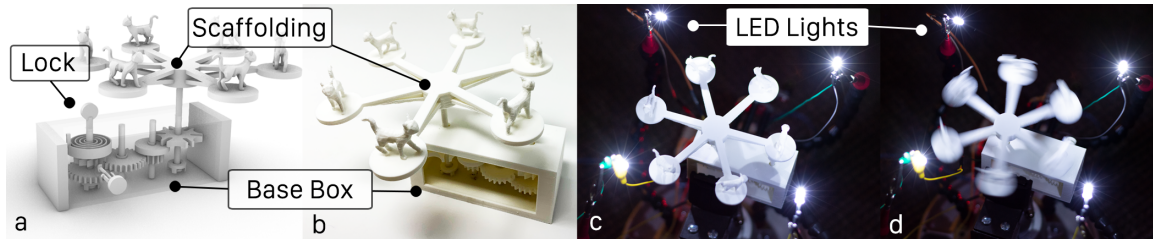


Figure 5.18: A 3D-printed zoetrope installation that is driven by an intermittent rotation kinetic unit embedded base box and animates a 3D walking cat: (a) the rendered model of the zoetrope, (b) the printed setup with programmable LED flashlights, and (c) the rotating zoetrope in motion.

5.6 Limitations

Kinergy customizes 3D-printable motion by embedding springs and kinematic mechanical components in a 3D-printed object. However, there are several limitations to this approach.

Geometry Complexity. The size of an object is limited to the minimum spring size and the minimal room for hosting printable kinematic parts in the transmission mechanism. This problem results in two limitations with our current tool: geometries with smaller sizes are not converted, and large gear ratios are not supported (currently, the gear ratio range is 1/5-5). As we have validated the feasibility of *Kinergy* with other multi-material 3D printing technologies such as PolyJet 3D printing, we plan to explore higher resolution printers as alternatives to mitigate both concerns. For example, smaller gears with finer gear teeth can be created for objects with smaller body volume. Another limitation of our approach is that the component engagements in the embedded kinetic unit require a compatible object topology. For example, in the continuous rotation kinetic unit with pressing as the energy-charging method, the rack-and-pinion mechanism that couples the helical spring and the geartrain must follow a strict moving path, which leads to an unchangeable spatial configuration. This problem could be further addressed by introducing an alternative yet complex transmission design. For example, we can use bevel gears, which have conically shaped tooth-bearing faces and rotate around non-parallel axes, in the geartrain to provide

more arrangement options for engaging gear teeth in a compact shape.

Printability and Robustness of Mechanical Parts. *Kinergy* benefits from the in-place printing and the predictable movements of kinematic components (*e.g.*, springs, gears, axles, joints); however, compared to industrial manufacturing methods such as casting and forging, 3D-printed mechanical kinematic parts are limited to the anisotropy and resolution of 3D printing. For example, 3D-printed springs are more robust and less brittle when printed perpendicular to the 3D printer's Z-direction. With the current setup, a printable gear has a minimum 1 module, which measures gear size by dividing the gear diameter by the number of teeth, and a minimal 0.3mm backlash. From our fabrication exploration, I found that tolerance gaps may be varied to applying *Kinergy* to different 3D printers and materials. I also found that objects printed with higher infill density (*e.g.*, 60% or above) are less brittle and more reliable. One solution is to use alternative material, and printing processes that produce parts with high tensile and endurance strength against loads but are less impacted by the anisotropicity of 3D printing. For example, 3D printing metal, a rising 3D printing industry, could be used to create more durable and sophisticated kinematic parts. The main goal of *Kinergy* is to convert the potential energy stored in the embedded spring into the output motion. However, energy reduces due to the friction caused by the relative movement between mating kinematic parts. To minimize friction, I used kinematic components that have point or curve contacts between moving parts and thus caused less friction [2] in the kinetic units. Additionally, I also explored methods to reduce friction. For example, friction was greatly diminished by adding lubricant such as grease or oil to the engaging points in the kinematic parts.

Energy-Releasing Triggers. The current lock mechanism in a kinetic unit allows only one way to release the stored energy in the spring, which prevents the customization of the interaction with the energy source and may result in onerous human operations to trigger the action. For example, to launch the pull-back car, the user must pull the latch, which may interfere with the car's movement. Additional sophisticated mechanical structures could offer a design space for customizing human-operated energy-releasing triggers. For example, a pressable bistable button may be more practical in the pull-back car example. Besides human-operated triggers for releasing energy, custom triggers can also be devised

to connect multiple kinetic units and thus execute a series of motions in a controllable sequence. In the future, I plan to investigate feasible mechanical designs that allow the user to control when and how the energy can be passed from one kinetic unit to the next. One promising solution is to use the tourbillon mechanism as a mechanical timer to bridge two kinetic units. For example, to create an automated door opener, a button with an embedded continuous translation kinetic unit receives the pressing from the user and passes the translation to a tourbillon-based timer, which extends a crank to rotate a concatenating door hinge with an embedded instant rotation kinetic unit after a few seconds.

User Interface Improvements. The current user interface of *Kinergy* provides a set of interactive controls for the end-user to add kinetic units to 3D models and parameterize added components for desired motion. Most controls ask the user to interact with abstract concepts such as speed and energy strength. One improvement for the user interface is to provide iconographical and easy-to-understand components for the end-user. For example, using the graphics of weight with labels to indicate the energy strength or providing a series of on-boarding tutorials to help the user understand and familiar with all the controls in the user interface. Furthermore, as *Kinergy* is applicable to other 3D printing approaches, I plan to incorporate the compatible tolerances and printing-related metadata such as material types and infill density for specialized printers and approaches in the tool. Furthermore, *Kinergy* can be extended with intelligent algorithms to optimize the generation of kinetic units for various 3D printing processes and materials.

5.7 Chapter Summary

In this chapter, I presented *Kinergy*, an interactive design tool that allows the user to create self-propelled motion for 3D printing with a set of parameterizable kinematic designs. First, I provided the background about the motion types for 3D printing. Then I introduced 3D printable kinetic units, which consist of embedded energy sources (either a helical or spiral spring), compliant locks, and transmission mechanisms, to enable the customization of 3D printable non-periodic and periodic motion. To embed these kinetic units into custom 3D models, I presented a design tool where the user, who is a 3D modeler but lacks an engineering background to create functional kinematic 3D models, parameterizes embedded

kinetic units to control the energy and motion characteristics. I then detailed the user interface of the design tool and the parameterization of energy and motion through custom user controls. Finally, I showcased how *Kinergy* supported the design and fabrication of 3D printable movements via a series of examples and discussed the improvements to our approach.

Chapter 6

**FLEXHAPTICS: CREATING CUSTOM HAPTIC INTERFACES
USING VARIOUS PLANAR COMPLIANT STRUCTURES**



Figure 6.1: Example applications made with *FlexHaptics* method: (a) a piano keyboard interface for touchscreen musical applications, (b) a VR controller attachment for bow shooting games, and (c) a joystick with a two-step button on the stick end.

This chapter describes the last project, *FlexHaptics* [65], highlighting how to leverage flat planar compliant structures (*e.g.*, spiral and zigzag beam structures) to create custom haptic input interfaces (RQ1) and how to control the force-deformation relationship in these structures by designing the structure geometries in a design editor (RQ2)¹. This project was led by Dr. Hongnan Lin from Georgia Institute of Technology, with assistance of tool developing from Yifan Li and me ², as well as feedback from Dr. Hyunjoo Oh, Dr. Wei Wang, and Dr. Clement Zheng. *FlexHaptics* extends spring-based 3D printable kinetic objects to support user input, which is different from the two 3D printable output behaviors enabled by *Ondulé* and *Kinergy*—deformation and motion.

Haptic feedback in tangible input interfaces is critical to enhancing user performance and engagement [140]. However, off-the-shelf components with predetermined and fixed haptic profiles are insufficient to satisfy an increasing need for sophisticated interaction design for various user scenarios. Customizing passive haptic inputs with fixed force-movement profiles

¹The full video demo of FlexHaptics: <https://youtu.be/GWIFYzo-zAYM>

²The code repository for FlexHaptics design editor: <https://github.com/hlin0101/FlexHaptics>

determined by mechanical mechanisms expands design opportunities for multiple interfaces. *FlexHaptics* aims to forward the computer-aided design and fabrication of haptic inputs with predictable force feedback by introducing a new method that leverages beam structures superior in their simple geometries and predictable haptic properties. This method supports inputs with eight mechanical module designs that provide predictable force feedback with embedded beam structures. Each unit generates a distinct haptic effect, including resistance, detent, or bounce, while traveling along a linear, rotary, ortho-planar linear movement path. The modules are planar and compact, therefore are easy to fabricate by 3D printing PLA, laser-cutting an acetal plastic (POM) sheet, or laser-cutting an acrylic sheet. The form factor also aligns with the construction of modern products, such as touchscreen devices, gamepads, and keyboards. Moreover, it satisfies the common need to arrange multiple inputs within a small space. To use these modules, we proposed two mixing operators to guide designers in composing the modules: parallel mixing generates an input with multiple haptic effects along a primitive path, and series mixing generates an input with a compound path.

To allow designers to use these modules to create custom haptic input interfaces, *FlexHaptics* also comprises an editor as a plug-in within Rhinoceros and Grasshopper to let designers explore module design according to desired force feedback. To implement the editor, first, we developed mathematical models quantifying the haptics-geometries relationship of each module. The models are in the format of linear regression analysis equations with an explanatory variable of a composite geometric component, informed by existing theories and models. Then we computed the coefficients and validated the models through finite element analysis (FEA) and experiments with modules fabricated with the three methods. Next, we adjusted the generation algorithms to address fabrication issues based on the experiment results. Finally, based on the experiment results, we developed the back-end of the editor with Grasshopper and Rhinocommon in C# and the front-end interface with Human UI. With the design editor and modules, we built six example applications that demonstrate a broad spectrum of use cases applying the proposed techniques and the tool, including two on-touchscreen interfaces of a slider control and a piano keyboard, a tactile low vision timer, VR game controllers, and a joystick combining with a two-step button (see some

applications in 6.1). In particular, highly conductive copper tapes can be attached to the deformable compliant structures to enable capacitive sensing for interactions.

My primary contributions to this project include providing FEA simulation support for validating *FlexHaptics* modules, developing the user interface for the design editor, and ideating the module designs and applications. In this chapter, I briefly describe the *FlexHaptics* modules and mixing operations, the design editor and the editor's implementation, and the six applications created with *FlexHaptics*.

6.1 *FlexHaptics* Modules and Mixing Operators

FlexHaptics approaches haptic input design by parameterizing two attributes: the movement path and the haptic effect. The movement path includes linear and rotary motions, and the haptic effect comprises resistance, detent, and bounce feedback patterns. As shown in Figure 6.2, resistance refers to a force with a steady magnitude and direction opposite to movement; detent refers to a force-displacement pattern where resistance increases and decreases within a short displacement; And bounce refers to a resisting force that decreases or increases as displacement decreases or increases.

By crossing the movement paths and haptic effects, *FlexHaptics* provides eight primitive modules, each of which supports a haptic effect along a movement path. The modules include linear resistance, linear detent, linear bounce, rotary resistance, rotary detent, and rotary bounce modules (Figure 6.2), and additionally, an ortho-planar linear bounce module that generates a bounce effect along an out-of-plane linear movement path and a planar bounce module whose mobile part moves free on the plane and exerts bounce effect.

6.1.1 *Resistance Feedback*

To achieve resistance feedback, two resistance modules were developed: a *linear resistance module* and a *rotary resistance module*. A resistance module consists of a deformable mobile part squeezed into and moving along a linear or rotary track. The raised-end cantilever embedded in the mobile part deflects and exerts a reaction force, which transfers to sliding or rotating resistance between the two parts with a specific friction coefficient.

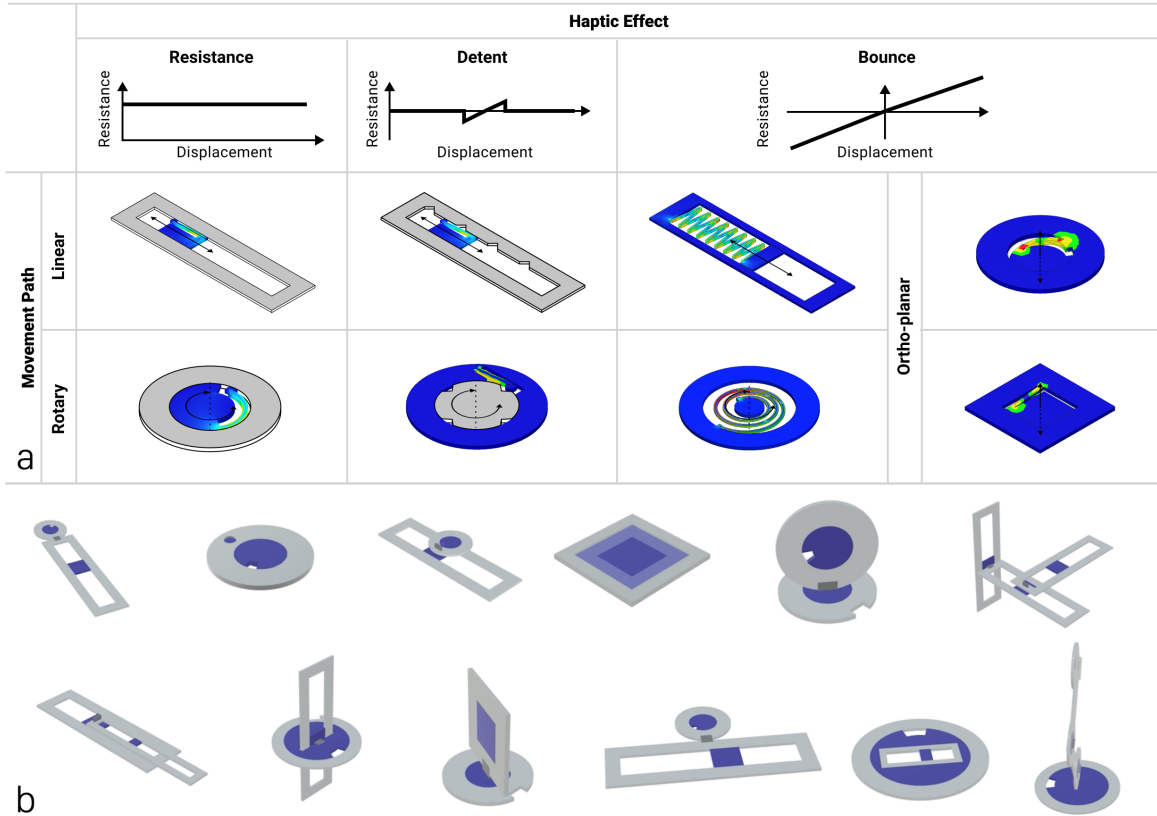


Figure 6.2: (a) A *FlexHaptics* module supports a haptic effect among resistance, detent, and bounce, along a linear or rotary path. The left six modules afford within-plane path, the two additional modules are designed for bounce effect along an out-of-plane linear path. Gray parts are rigid; colored parts are compliant, and color changing from blue to green, and to red indicates increasing stress levels. (b) All the modules can be mixed for complex haptic feedback and moving paths.

Linear resistance modules. A linear resistance module provides a constant force resistance to sliding (Figure 6.3). The deformed beam yields a reaction force F_{LR-r} predictable with l , b , h , and free end displacement when displacement is small according to *Euler-Bernoulli* theory, and displacement equals r_t (Eq. 6.1). Normal force N between the two parts can be calculated from the reaction force, according to the mechanical equilibrium equations of the tip and slider (Eq. 6.2). Force feedback of a linear resistant module F_{LR} equals the sum of sliding resistance, which is the product of normal force and friction coefficient μ decided by material property (Eq. 6.3). In summary, force feedback from a

linear resistance module of a specific material is predictable from the module geometry, as shown by substituting Eq. 6.1 and 6.2 into 6.3.

$$F_{LR-r} = A \frac{r_t b h^3}{l^3} + B \quad (6.1)$$

$$N = 2F_{LR-r} \quad (6.2)$$

$$F_{LR} = \mu N \quad (6.3)$$

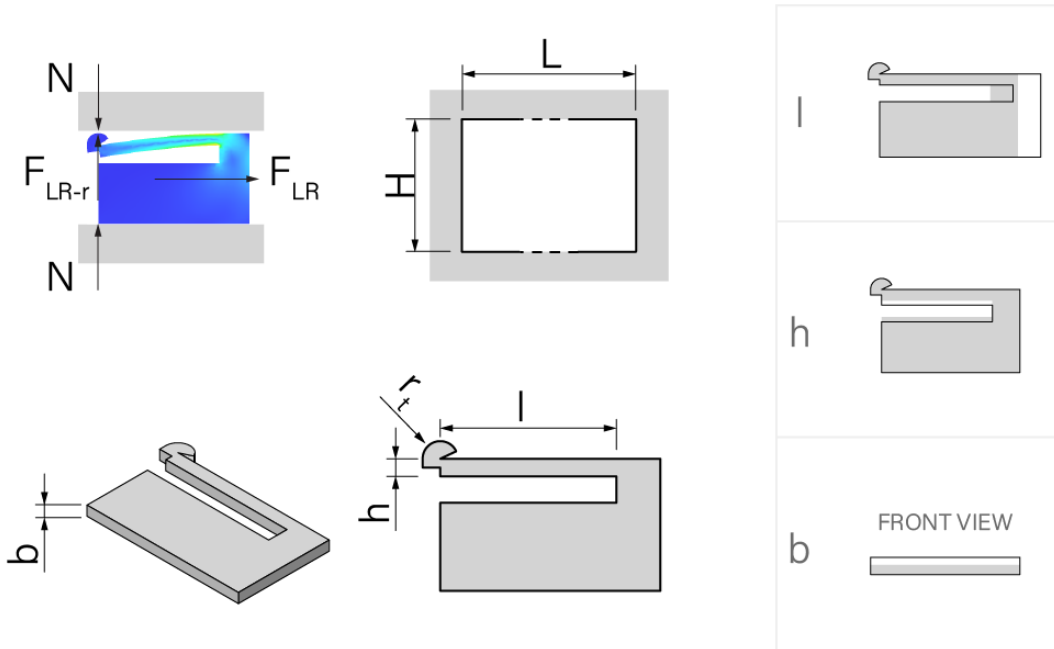


Figure 6.3: Linear resistance module. It comprises a flexure slidable along a linear track. Its force feedback is adjusted with beam length l , thickness h , and width b .

Rotary resistance modules. A rotary resistance module provides a constant torque resistance to rotating (Figure 6.4). The deformed beam exerts a reaction force predictable with r , a , b , h , and displacement according to *Castigliano's* theorem, and displacement equals r_t (Eq. 6.4). The normal force between the two parts can be calculated from the reaction force, according to the mechanical equilibrium equations of the tip and rotor (Eq.

6.5). The feedback torque of a rotary resistant module T_{RR} equals the sum of resistant torque, which is the product of normal force, friction coefficient, and R (Eq. 6.5). In summary, the feedback torque from a rotary resistance module on certain material is predictable from the module geometry and material, as shown by substituting Eq. 6.4 and 6.5 into 6.6.

$$F_{RR-r} = A \frac{r_t b h^3}{r^3 (2a - \sin 2a)} + B \tag{6.4}$$

$$N = 2F_{LR-r} \tag{6.5}$$

$$T_{RR} = \mu N R \tag{6.6}$$

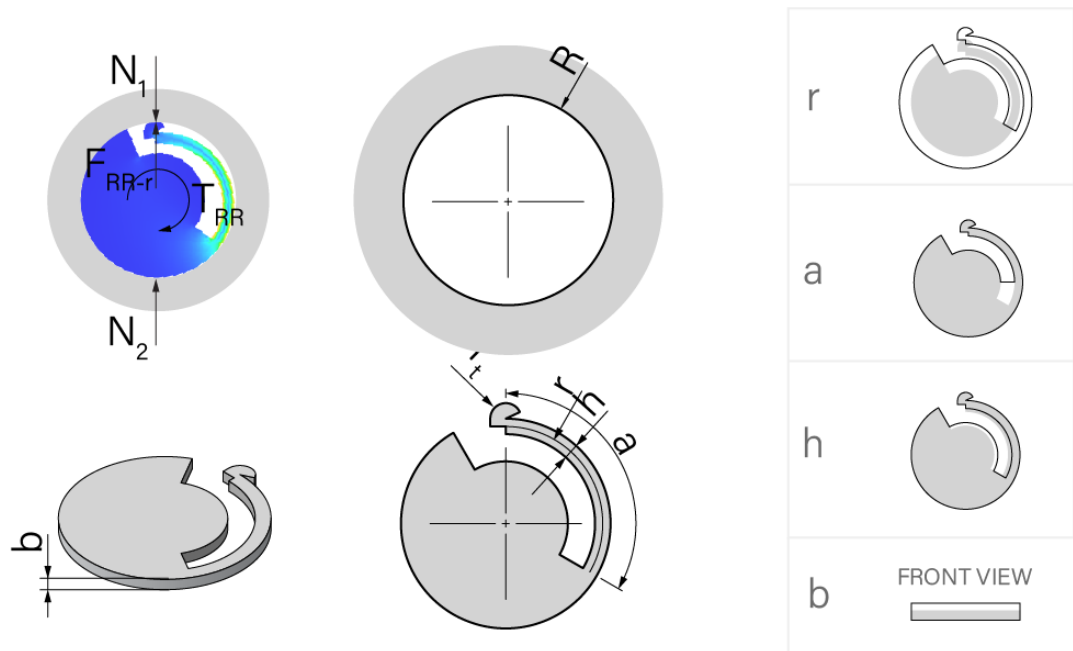


Figure 6.4: Rotary resistance module. It consists of a flexure rotatable within a ring. Its force feedback is adjusted with beam radius a , radius r , thickness h , and width b .

6.1.2 Detent Feedback

To achieve detent feedback, we also developed two detent modules. A detent module consists of a mobile part moving along a linear or rotary track with notches (Figure 6.5). The mobile parts employ the same beam structure of linear resistance modules, with a raised tip conforming to the notches. Friction between a mobile part and track is removed by adding lubricant, and force feedback of sliding or rotating equals the tangential component of normal force between the two parts.

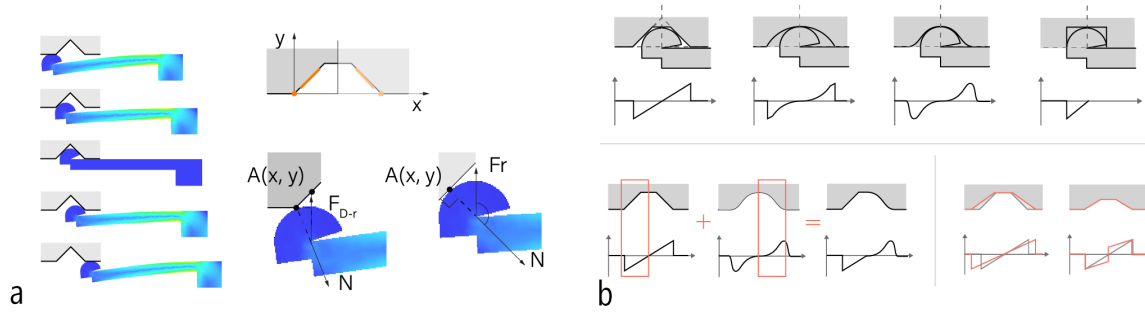


Figure 6.5: Detent modules. A linear or rotary detent module employs the same beam geometries as linear resistance module and adapts notches to contact surface. (a) As the beam moving across a notch, force feedback is determined by the notch and beam geometry. (b) We present four symmetrical notch signatures and force-displacement curves. Mixing a left and a right side of them generates another 12 detent profiles. Force feedback from a notch can be adjusted by scaling it along its width or depth direction.

Given the geometry of a detent module, what is known is the notch profile $y = f(x)$, where x and y are the horizontal and vertical position of a point on the notch, the notch slope is $f'(x)$, and the beam stiffness k predictable from b , l , and h (Eq.6.1), and tip radius r_t . What can be calculated is beam deflection δ (Eq.6.7 or 6.8) and reaction force (Eq.6.9) when the beam moves along the notch. And the force feedback during the movement is the horizontal component of the reaction force (Eq.6.10).

When the contact point is derivable (Figure 6.5a right),

$$\delta = \sqrt{r_t^2 - x^2} \quad (6.7)$$

When the contact point is not derivable (Figure 6.5a left),

$$\delta = \text{tr} \left(1 - \frac{1}{\sqrt{1 + f'(x)^2}} \right) \left(-f(x) \right) \quad (6.8)$$

$$F_{D-r} = k\delta \quad (6.9)$$

$$F_D = F_{D-r}f'(x) \quad (6.10)$$

Various detent effects can be designed by adjusting notch profiles, beam geometries, and notch distributions. A notch profile can be created by selecting a left side and a right side from the four preset notch profiles: 1) constant slope, 2) increasing slope, 3) increasing then decreasing slope, and 4) locking. A notch can be scaled along or perpendicular to the movement direction, influencing the detent effect scope and sharpness. Adjusting the beam increase or decrease force feedback along with the notch consistently.

6.1.3 Bounce Feedback

Finally, to achieve bounce feedback, we developed four modules: linear bounce module, rotary bounce module, ortho-planar linear bounce module, and ortho-planar rotary module. Bounce modules exert a restoring force (F) toward the equilibrium and are proportional to displacement when the mobile part is moved away from the neutral position within a range. Furthermore, bounce coefficients can be adjusted by altering geometric parameters.

Linear bounce module. A linear bounce module provides a resistance force proportional to displacement when stretched or squeezed within a range (Figure 6.6a). A linear bounce module is constructed with a series of beams. The bounce coefficient $k_{LB-unit}$ of each unit can be predicted with l , b , and h (Eq.6.10), and that of the whole module k_{LB} can be further calculated with the number of units based on series spring formulas (Eq.6.11).

$$k_{LB-unit} = A \frac{bh^3}{l^3} + B \quad (6.11)$$

$$k_{LB} = \frac{k_{LB-unit}}{n} \quad (6.12)$$

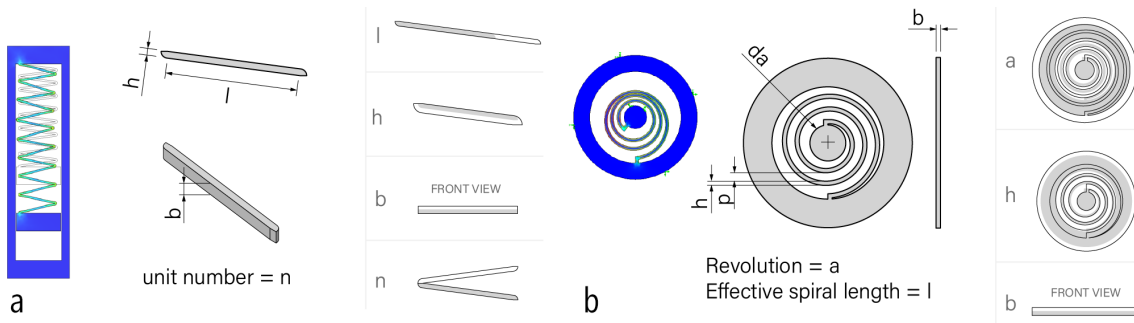


Figure 6.6: Bounce modules. (a) A linear bounce module can be stretched or compressed, its stiffness can be adjusted by beam length l , thickness h , and width b , and unit number n . (b) A rotary bounce module can be rotated clockwise or counterclockwise, its stiffness can be adjusted by spiral radian a and wire thickness h and width.

Rotary bounce module. A rotary bounce module allows the arbor to rotate and exerts reaction torque proportional to the rotating angle (Figure 6.6b). The design of the spring was based on the standard Archimedean spiral defined by da , h , p , and a , from which its effective length l can be calculated (Eq.6.13, 6.14, and 6.15). And the bounce coefficient can be calculated with b , h , and l according to spiral spring theory (Eq.6.16).

$$l = -\frac{1}{2y} \left[x\sqrt{x^2 + y^2} + y^2 \ln \left(x + \sqrt{x^2 + y^2} \right) \left(- (x + ya) \sqrt{x^2 + y^2 + 2xya + y^2a^2} - y^2 \ln \left(x^2 + y^2 + 2xya + b^2a^2 \right) \right] \quad (6.13)$$

$$x = \frac{da + h}{2}, \text{ where } d = 10 \quad (6.14)$$

$$y = \frac{p}{2\pi}, \text{ where } p = 2h \quad (6.15)$$

$$k_{RB} = A \frac{bh^3}{l} + B \quad (6.16)$$

Ortho-planar bounce modules. Ortho-planar linear bounce modules provide the mobile platform moving out of the base plane with a resistance force proportional to displacement (Figure 6.7). They are constructed with a straight (Figure 6.7a) or round (Figure

6.7b) beam following the mobile platform shape. Bounce coefficients of straight beams can be calculated with l , b , h . Bounce coefficient of round beams can be predicted with l (i.e., a and r), b , h .

$$k_{OLBS} = A \frac{bh^3}{l^3} + B \quad (6.17)$$

$$k_{OLBR} = A \frac{bh^3}{(ra)^3} + B \quad (6.18)$$

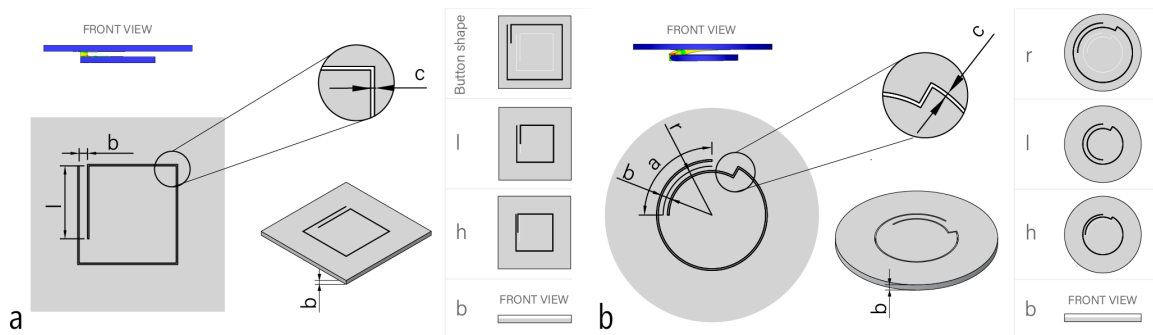


Figure 6.7: Bounce ortho-planar modules. (a) A straight-beam ortho-planar bounce module can be adjusted with beam length l , thickness h and width b . (b) A curve-beam ortho-planar bounce module can be adjusted with beam radius r , radian a , and beam thickness h and width b .

6.1.4 Mixing Operations

Multiple modules can be combined into composite inputs through two mixing strategies: mixing in parallel and mixing in series (Figure 6.8). Mixing in parallel aligns and respectively bonds the mobile and static parts of two or more modules with the same movement path, producing a compound input with multiple haptic effects along the single path, and mixing in series bonds the base of one module or composite input to the mobile part of another, and so on and so forth. The resulted inputs have a complex movement path and maintain the component haptic effect(s) along each component path.

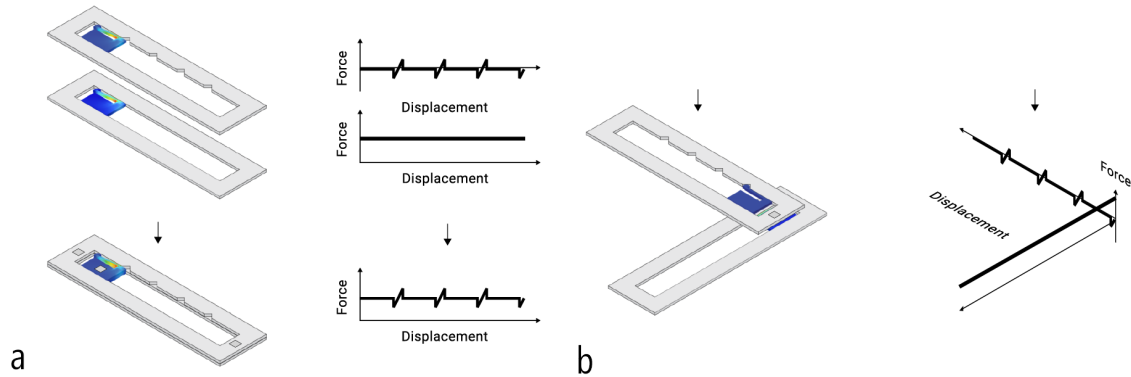


Figure 6.8: Two mixing operators for *FlexHaptics* modules. (a) Mixing in parallel aligns modules with the same movement path and bonds the mobile and static parts together respectively, resulting in an interface with the same movement path and a compound haptic effect. (b) Mixing in series uses multiple modules with different movement paths, and bonds the static part of one module to the mobile part of another module, producing an interface with a complex movement path.

6.2 Technical Evaluation

To identify effects from fabrications and inform adjustments to *FlexHaptics* techniques, we evaluated how well the mathematical models predict modules of FEA simulations and modules fabricated with the three methods (laser cutting with acrylic and POM and 3D printing with PLA). FEA simulation allowed for investigating ideal modules where the sizes are precise, and the material properties are the same as those provided by the manufacturers. In the technical evaluation, we simulated and measured the reaction force of resistance modules and stiffness of detent and bounce modules with varying geometric parameters. The prediction method predicted the reaction force or stiffness of FEA modules with high goodness-of-fit (averaged $R^2 = 0.99836$), and fabricated modules with less goodness-of-fit (averaged $R^2_{Delrin} = 0.95341$, averaged $R^2_{acrylic} = 0.79088$, averaged $R^2_{PLA} = 0.97723$). Different fabrication methods introduce dimensional errors and change material properties. Besides the beams' reaction, the force feedback of a module was affected by other factors like friction and track stiffness. We identified four approaches to avoid or mitigate the errors. The first is to avoid setting small values of h and b . The second is to prioritize adjusting l

values instead of b or h to meet haptic values. The third is to fabricate a series of models with l values varying around the value calculated for a desired force or stiffness value so that one of the prototypes can provide the desired haptics. Lastly, we noted avoiding using acrylic for linear bounce modules because of the high fracture risk.

6.3 *FlexHaptics Design Editor*

Informed by the module design and technical evaluation, we developed an open-source design editor to make the *FlexHaptics* technique available to designers. To generate a module in *FlexHaptics* editor (Figure 6.9), designers need to go through the following steps:

1. Choose a module type.
2. Select a fabrication method and material.
3. Set desired haptic values. The editor generates geometry in the 3D viewport and calculates its force feedback with error range.
4. Explore other possible geometries. Because different parameter combinations can produce the same force feedback, this step allows designers to adjust a module freely while comparing its force feedback to that set in Step 3.
5. Export a final design in STL or SVG format.

6.3.1 *Implementation*

The backend of *FlexHaptics* editor is implemented with Grasshopper and Rhinocommon in C#, and the frontend interface is developed using Human UI.

For Step 1, the user interface provides a list of buttons for the user to select the desired *FlexHaptics* module. For Step 2, the editor uses pre-defined parameters for different fabrication techniques—for example, the printing tolerance for 3D printing with PLA.

For Step 3, the editor takes in the parameter values input by the user in the user interface and generates the compliant structure geometries based on the equations shown

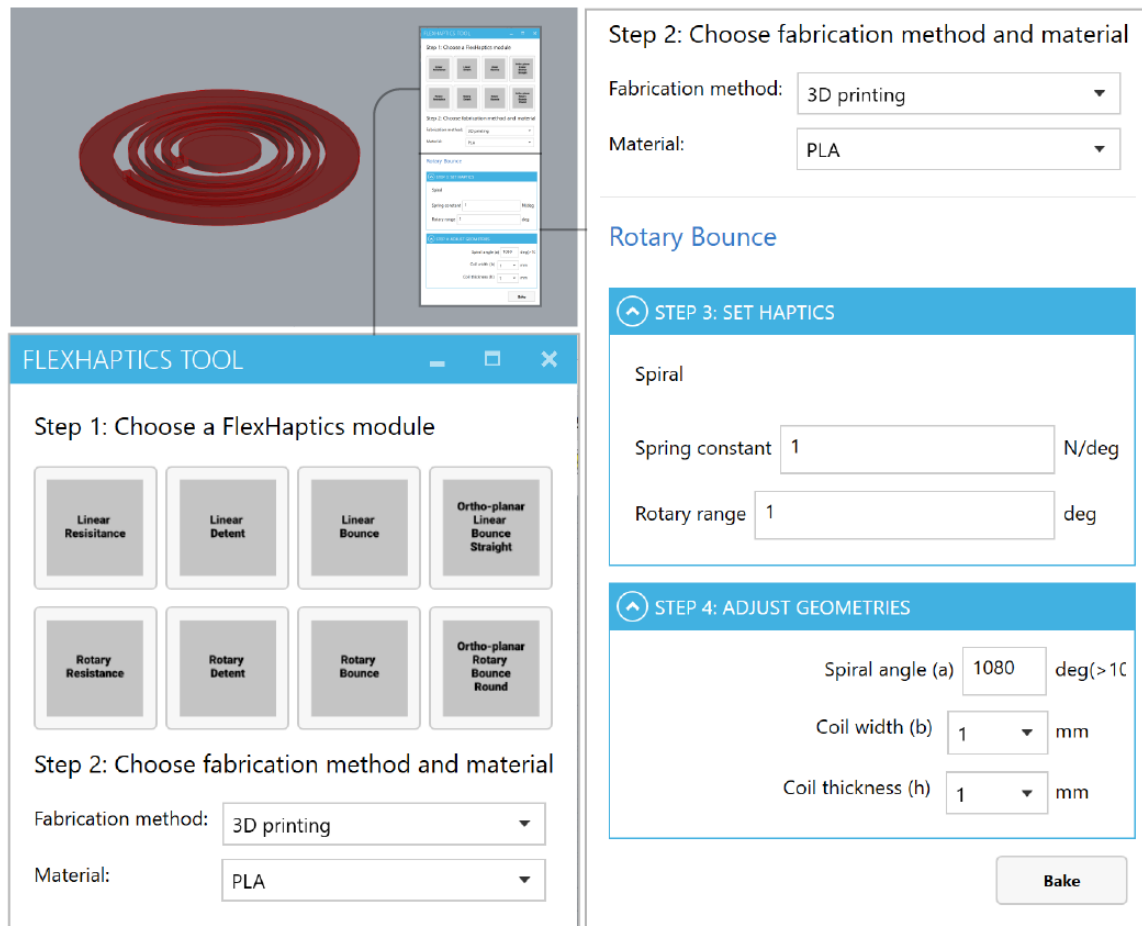


Figure 6.9: *FlexHaptics* user interface consists of a *FlexHaptics* tool panel and module preview in Rhinoceros environment.

above. The 3D model of the compliant structure is created in real-time in the 3D editing Rhino scene. For Step 4, given geometric parameters, haptic values are calculated using the above equations. For instance, linear resistance module, on receiving user input haptic value in Step 3, the algorithm first calculates the reaction force with corresponding equations (Eq 6.2 and 6.3). Then to calculate geometries from reaction force with Equation 6.1, the algorithm starts by setting b and h at the smallest value to calculate l , then examine if l is at least ten times as long as r_t (to meet small deformation condition). If not, the algorithm will increase h and b step by step and repeat calculating and evaluating l . If yes,

the algorithm will use the values to generate geometry. A similar process applies to the other modules. Once deciding on the compliant part geometry, the algorithm will adjust the other component for the two-part modules to maintain a fit clearance and can offset the geometries for laser cutting and chamfered for 3D printing. All the updates are rendered in real-time.

6.4 Haptic Interface Examples Created with *FlexHaptics*

We developed six example applications to validate the proposed technique that covers different modules, mixing operators, fabrication methods, and materials. These applications were created in three categories according to application environments: haptic layers above graphics on touchscreens, passive haptic proxies in VR, and haptic controls with microcontrollers.

6.4.1 Haptic Layers above Graphics on Touchscreens

We demonstrate two examples made by laser-cutting acrylic sheets and attaching copper tapes to align with graphics and transit user touches on touchscreens.

Haptic control panel for a painting application. The haptic control panel for a painting application, Procreate, is proposed to reduce divided attention caused between canvas and toolbar (Figure 6.10a). The transparent haptic layer overlaid on the graphical interface employs a linear resistance module for changing brush size, and a linear detent module notched at preferred values for adjusting opacity. Knobs inserted into the sliders are wrapped by copper tapes to transmit user touches to a touchscreen.

Piano keyboard. We built a piano keyboard interface for a touchscreen music app (Figure 6.10b) to improve user performance. Similarly, the keyboard is made from an acrylic sheet and copper tapes around each key. In addition, we highlighted that ortho-planar linear bounce modules could resemble such keyboards by being shaped and collocated.

6.4.2 *Passive Haptic Proxies in VR*

When using passive props for VR interaction, a key could be to use less physical materials to simulate more virtual objects. *FlexHaptics* could be a solution because its compactness increases the number of proxies containable within a reasonable space.

VR controller attachment for bow-shooting games. This VR controller attachment system for bow-shooting games (Figure 6.10c). Players choose a proxy at the preferred level of resistance to use a virtual bow, simply plug its base end into the socket on one controller, and draw or release the free end with the other controller. A bow proxy consisting of a laser-cut POM linear bounce module sandwiched by acrylic faces simulates increasing resistance while opening a bow and retracting effects while releasing an arrow. The position and orientation of a proxy can be drawn from those of the associated controller.

String-based wearable haptic device. We demonstrate a haptic device worn between the wrist and fingertips, which provided haptic feedback to different gesture interactions in VR (Figure 6.10d). Our device comprised multiple stackable string-retracting units, each of which leverages a rotary bounce module for retracting and another rotary module for more haptic effects. We made the modules by PLA, as it is easier to integrate other extruded structures, like bobbins and tongue joints. Thanks to the compact form factors of the *FlexHaptics* modules, a stack of three units is as tiny as a smartwatch.

6.4.3 *Haptic Controls with Microcontrollers*

Drawing on research and projects on paper circuits, we built most circuits with copper tapes and conductive ink, manually or using a cutting machine.

Tactile timer input. We presented a tactile timer input for low-vision people (Figure 6.10e). Each dial consists of a rotary resistance and detent module. The hour, minute, and second dials present diminishing resistances and different detent densities. Starting and canceling buttons were made with ortho-planar modules. This prototype highlighted a compact design enabled by nesting the three dials in the same layer.

Haptic controls with complex movement paths. The joystick designed for a shooting game can control shooting direction and switch between single or scattering shots

(Figure 6.10f). It allowed rotation in horizontal and vertical planes by serial-mixing two mutually perpendicular rotary bounce modules. The button on the handle could activate single or scattering shooting mode under light or hard press. It employed in-series mixing of two ortho-planar bounce modules with different bounce constants. The modules are made by laser-cutting POM sheets. This example demonstrates a complex 3D movement path by mixing modules in different planes.



Figure 6.10: Application examples: (a) a slider input interface for touchscreen painting applications, (b) a piano keyboard interface for touchscreen musical applications, (c) a VR controller attachment for bow shooting games, (d) a string-based wearable haptic device, (e) a tactile low vision timer, and (f) a joystick with a two-step button on the stick end.

6.5 Limitations

As *FlexHaptics* provides a set of enabling techniques and a design tool for designers to create custom haptic input interfaces, limitations are enumerated as follows:

Fabrication error, material fatigue and creep, external friction. The current mathematical models are limited due to three causes of differences between actual and predicted haptic effects of the *FlexHaptics* interface. The first is fabrication errors, as indicated by the less goodness of fit for the fabricated modules. Designers can make up by making a range of models containing a desired one. The second is material fatigue and creep, as with most techniques. The detent and bounce modules in interaction are under cyclic loading and thus will get weaker after extensive use. The resistance and detent modules are influenced by persistent mechanical stresses even not in usage, thus will exhibit dimmer haptic effects after a long term of preservation. Designers need to replace exceedingly used or old ones to maintain haptic preciseness. These two factors can be addressed by exploring more fabrication methods and materials. Lastly, the actual haptic effects of an interface come from the modules and friction between the modules and external objects (*e.g.*, circuit layers, housing structures, or touchscreens), which is unpredictable by the *FlexHaptics* editor. Designers can iterate modules to counteract the external friction for satisfying results. There are many ways to avoid external frictions, for example, minimizing surface areas in contact with external objects or maintaining gaps between the modules and external objects via housing structures.

Supplementary structures. The method does not assist in designing two types of supplementary structures often necessary for *FlexHaptics*-enabled interfaces. The first type helps constrain mobile parts to designed movement paths, *e.g.*, preventing those in resistance, detent, and linear and rotary bounce modules from moving out of tracks and guiding that in a rotary bounce module to rotary instead of translational movements. The second is bonding parts in mixing modules or bonding a module to other structures. Among tested techniques, like 3D printing different parts as one and gluing, tongue-and-hole jointing stands out as it is strong for connection and flexible for replacement. The design of supplementary structures varied to adapt to different applications. It was not very diffi-

cult but certainly complicated the overall design process. In the future, these additional structures can be automatically incorporated into the design editor and generated by the tool.

Broadening materials. *FlexHaptics* editor currently supports limited material choices. In future work, we plan to widen the available material selections based on the mathematical models—for instance, SLA resins and metals for stronger, compacter, and more accurate modules. We also plan to adapt elastic materials (*e.g.*, TPU) and even programmable filaments to expand the design and fabrication space of haptic input interfaces. Taking resistance modules as an example, applying multiple materials with different friction coefficients to the track boundary produces a module with varying resistance along the path. Finally, stimuli-responsive material could enable *FlexHaptics* modules with adaptive haptic feedback. For instance, applying stiffness-changing material to the flexures could produce modules with adaptive resistance, bounce coefficient, and detent magnitude. Likewise, applying shape-changing material to the notches could create detent modules with adaptive feedback profiles and distribution.

Active force feedback and vibration feedback. This method is limited to passive force feedback generated by the module design. Due to the passive nature, *FlexHaptic*-enabled interfaces cannot automatically perform tasks (like the timing task in the timer example), proactively initiate touches, or spontaneously adjust profiles. Future work could investigate ways to actuate *FlexHaptics* modules while maintaining their compactness, like using smart materials. Besides, force feedback, vibration is also essential as they widely exist in natural and artificial interactions. We excel at perceiving with numerous receptors embedded in our skin and other haptic organs. Interaction with resistance and detent modules involves vibration, which can simulate human sensitivity by adjusting beam stiffness and track texture. It is possible to introduce texture by laser-cutting a smooth line with low-power settings on extruded acrylic sheets and designing notch profiles and distributions. In addition, supplement vibration modules will be added to *FlexHaptics*.

User-centered evaluation. While we ran a thorough technical evaluation with the modules, a formal user evaluation of the *FlexHaptics* techniques and editor is needed to improve the system’s usability. In the future, we will investigate whether designers can

adapt and find the proposed modules and mixing operators useful in their design processes and identify areas to improve further.

6.6 Chapter Summary

In this chapter, I presented *FlexHaptics*, a computational design method to create haptic input interfaces with custom feedback using a low-cost 3d printer or a laser cutter. *FlexHaptics* editor comprises eight primitive modules that exert a haptic effect, *i.e.*, resistance, detent, and bounce, along a movement path, *i.e.*, linear, rotary, ortho-planar linear, and planar. Each *FlexHaptics* module supports adjustable haptic effects via module geometries. With *FlexHaptics*, designers can create haptic interfaces that provide custom force feedback. This approach also worked with externally added conductive tapes to enable sensing. Finally, I also discussed the limitations of this work and future improvements.

Chapter 7

CONCLUSIONS AND FUTURE WORK

The goal of this dissertation has been to design, develop, and evaluate 3D printable spring-based mechanisms to promote the fabrication of kinetic objects for interactivity. My approach was threefold: (i) to draw upon mechanical engineering, physics, trends of making kinetic 3D designs in the maker community, and past fabrication research to inform the use of mechanical springs in creating kinetic objects and supporting interaction; (ii) to design and evaluate parametric spring-based mechanisms that utilize mechanical springs and kinematic elements for desired kinetic behaviors including deformation, actuation, and sensing through structural form changes; and (iii) to develop and evaluate interactive parametric design tools for end-users to create custom 3D printable kinetic models with embedded spring-based mechanisms for a variety of applications.

In this chapter, I would like to summarize the major contributions of this dissertation and discuss the directions I would like to explore in my future research.

7.1 Contributions

In this section, I restate the four contributions listed in the introduction chapter and summarize how each of the contributions was achieved. These contributions provide cohesive support for the dissertation statement:

We can design 3D modeling and printing techniques to embed and control parametric spring-based mechanisms into 3D-printable objects, which enables a new suite of applications for 3D printing.

7.1.1 Making Practices of 3D Printable Kinetic Designs

This dissertation first contributes a large-scale analysis that studies how to make 3D printable kinetic objects within the maker community and uncovers challenges and opportunities

for using springs in 3D printable movable objects (Chapter 3). This study examined what 3D printable kinetic things (3D printable mechanisms and behaviors) were created by makers, what design techniques and tools were used for creating kinetic 3D designs, and what challenges and opportunities for making kinetic 3D models? This study’s findings showed that most kinetic 3D designs were made with 3D printable mechanisms, including the two most commonly used kinematic elements—joints and hinges. Combining 3D-printed parts with external hardware and using elastic printing materials generate kinetic 3D models. Most 3D objects were created to achieve rotating, bending, articulation, translating, twisting, and compressing movements. Different mechanisms could be used to achieve the same output kinetic behavior, and various mechanisms were sometimes combined for a more complex output behavior. These behaviors were primarily found in toys and gaming gadgets. The study also revealed that most makers were likely to reuse or remake existing models for new applications, which indicated a high barrier to designing kinematic and functional 3D parts. Finally, printing kinematic mechanisms in place is challenging because the printing resolution and tolerance depend primarily on the 3D printer and printing settings. This challenge also leads to a lot of manual post-print assembly, which is troublesome and error-prone.

The findings of this study not only provide the status quo of making 3D printable kinetic objects within the maker community. I also related those results with fabrication research and identified commonalities and differences. While this dissertation pursued one direction informed by this study, I believe the results of this study have more value in inspiring other design and fabrication techniques for creating kinetic 3D-printed objects for I/O. For example, alternative prototyping techniques such as silicone casting are used for making interactive 3D-printed objects [31]. Therefore, the contribution of this study is beyond this dissertation and can provoke many interesting implications and ideas to promote the field of kinetic fabrication.

7.1.2 *Spring-Based Mechanisms for I/O Behaviors*

In this dissertation, I developed these design techniques to answer *what kinetic 3D printable objects can be created using spring-based mechanisms*. I explored (i) creating 3D-printed ob-

jects with controllable deformation behaviors by embedding parameterizable helical springs and joints, (ii) enabling 3D-printed models to perform controllable movements with embedded self-contained energy sources using helical or spiral springs together with other kinematic elements like gears, and (iii) creating custom haptic input interfaces with flat planar compliant structures. To use the springs as the core components, I promoted spring principles and theories from physics and mechanical engineering. I validated the feasibility of mechanical springs for 3D printing. These spring-based mechanisms were developed to showcase that 3D printable springs could help create expressive kinetic devices for interaction. As a result, this dissertation pushed the boundaries of personal fabrication applications with these spring-based mechanisms.

Combining helical springs and joints for controllable deformation behaviors.

Ondulé, described in Chapter 4, presented a set of novel spring and joint-based design techniques to control spring deformation behaviors. This contribution was achieved by controlling the geometric parameters of 3D printable helical springs and combining helical springs with various joint constraints. Furthermore, since the springs and joints were constructed for in-place printing with a dual-material 3D printer, the end-users can easily create and fabricate desired deformation behaviors without troubles caused by the manual assembly.

Regulating potential energy stored in springs for desired output motion. *Kinergy*, described in Chapter 5, used 3D printable helical and spiral springs as energy motors and translated stored energy into custom output motion behaviors via various kinematic components such as geartrain. This contribution was achieved by combining springs with different kinematic elements for specific motion behavior while parameterizing all these kinematic parts for custom motion characteristics such as speed. Again, to reduce human work for assembling mechanical components, all parts could be printed in place with multi-material 3D printers.

Creating haptic interfaces enabled by deformable planar compliant structures. *FlexHaptics*, described in Chapter 6, presented how to use planar compliant structures to simulate force feedback for various custom haptic interfaces. This contribution was achieved by customizing a set of haptic modules, which were made of parameterizable flat

beam structures (*e.g.*, spiral and zigzag beam structures). These *FlexHaptics* modules can be used as an individual element to provide haptic feedback along with a specific path or combined to offer richer force feedback following compound movement paths. Furthermore, the haptic interfaces created with *FlexHaptics* can be fabricated with home 3D printers or laser cutters.

Although these spring-based mechanisms were created for different interactions and purposes, I think the approach can be potentially generalized to other mechanism-based fabrication research. For example, we can explore the parametric design space of the mechanism of interest and exploit these parameters to control the mechanical properties of different behaviors. Therefore, I proposed and developed spring-based mechanisms to answer my research questions above and took a mechanical approach to address HCI problems.

7.1.3 *Design Tools for Spring-Based Kinetic Fabrication*

This dissertation also contributes a set of design tools that allow the end-user to utilize spring-based mechanisms in 3D models for desired kinetic behaviors. These tools share the following commonalities: (i) supporting the design of desired 3D printable behaviors using in-place kinetic spring-based mechanisms; (ii) allowing the end-user to customize desired behaviors by parameterizing embedded mechanisms in the model; and finally, (iii) showing the resulting behaviors through a preview for the end-user to validate. This contribution was achieved by abstracting the underlying parametric design of spring-based mechanisms from the end-user in those tools and providing user-friendly interfaces for the end-user to design, control, and preview the spring-based mechanisms embedded in 3D models. I believe these characteristics stay valid and are essential for other types of interactive design tools for 3D modeling and fabrication.

7.1.4 *Applications Enabled by Kinetic 3D Printable Objects*

The last contribution of this dissertation is the applications enabled by the developed 3D printable spring-based mechanisms and design tools. These applications, created with the proposed mechanisms and tools, captured different aspects of the spring-based mechanisms,

e.g., different spring and joint designs for a variety range of deformation behaviors in *Ondulé*, and application domains as many as possible such as accessibility and interactive I/O devices. While most of the created applications were still within the context of rapid prototyping, I envision that more robust, reliable, and even scalable applications can be made using the same set of these spring-based mechanisms and tools but with advanced 3D printers, *e.g.*, metal 3D printers or high-resolution PolyJet printers.

7.2 Directions for Future Research

In this section, I would like to enumerate the limitations of the approaches presented in this dissertation in three categories: fabrication, design and control, and application. Further, I also discuss the following steps to address these issues in the future.

7.2.1 Fabrication Limitations and Future Work

While this dissertation presents spring-based mechanisms that are 3D printable in place on consumer-grade FDM 3D printers, the performance of the kinetic behaviors that can be achieved is still limited by 3D modeling and fabrication constraints.

First, unlike metamaterials or microstructures with minimal basic unit designs, springs, the crucial part of my approach, usually come in a bigger size, especially when printed on desktop 3D printers. As a result, when a spring-based mechanism is embedded into 3D objects, the 3D model should be large enough to house the mechanism. For example, the minimal printable spring wire thickness supported by *Ondulé* was 1.6mm, and the printing tolerance was 0.4mm, which meant the minimum size of a printable helical spring was 4.0mm wide. Similarly, *Kinergy* needed more room if other kinematic elements were also printed to engage with the spring. The size of the printed spring-based mechanisms could be further reduced with a higher resolution printer or alternative printing method (*e.g.*, SLA with resin). The tradeoff is the cost and accessibility of those alternative 3D printers).

Second, since many parts were printed in place to ease the post-print process and support the movement, printing tolerance was challenging to determine case by case, and the friction between inter-engaged parts would influence the kinematic movement. The printing tolerance is primarily determined by the printing resolution and printing orientation. For

example, if printing with 0.2mm layer height on an Ultimaker 3D printer, the minimum printing tolerance is 0.6mm horizontally but decreases to 0.4mm in the vertical direction (along the Z-axis). The mechanical experiments for *Ondulé* also showed how printing settings such as infill density and printing orientation influenced the mechanical performance of the printed helical spring. While the effect of friction could be neglected in the joints for *Ondulé*, the friction caused between the engaged kinematic components in *Kinergy*, such as the crank and slotted levers and Scotch yoke, greatly affected the motion behaviors and dissipated the energy quickly. To determine the best printing tolerance for a specific model, we can collect a large number of experiment sample data that record most of the printing tolerance values for a 3D part to be printed with various printing conditions. This way, the recorded data list can be provided when the 3D model is sliced, and the printing tolerance will be decided during the slicing process. To diminish the influence of friction, we can apply lubricant to the movable printed parts or use an alternative 3D printing method such as SLA or PolyJet.

Finally, all the parts were printed on desktop 3D printers with plastics and used FDM 3D printing methods (except for the last two projects that used advanced 3D printers for sensing applications). As a result, the robustness of the printed spring-based mechanisms was determined by the printing settings, including infill density, infill patterns, layer height, and printing orientation. For example, in *Ondulé*, printing orientation will affect the spring's E and G , where 45° results in minimum values for both and 90° yields the highest. When multiple springs with varied orientations need to be printed at once, finding an ideal printing orientation that works for all springs would be challenging. If the printing conditions were not suitable for the spring, the printed spring became brittle and easy to break. One possible solution to produce robust and reliable prints is to use alternative materials and printing processes that produce parts with high tensile and endurance strength against loads but are less impacted by the anisotropy of 3D printing. Alternatively, a 5-DOF 3D printing method can always repose the 3D model to ensure the spring can be aligned appropriately for the best result.

7.2.2 *Design and Control Limitations and Future Work*

The design tools presented in this dissertation did three primary things: (i) abstracting the underlying construction of complex spring-based mechanisms from the end-user, (ii) converting high-level user input (*e.g.*, parameter controls) into geometric models, and (iii) displaying the resulting 3D models to provide user awareness of the model updates. However, many parts of the tools could be improved to advance user design and control of the kinetic model with embedded spring-based mechanisms. First, these tools lack a mechanism to notify the end-user what would fail and where errors could occur during the user operations with the tool. As a result, the user might not be aware of the wrong inputs, and the tool might not function as the user expected. Second, although most of the parameter inputs were provided with interactive user controls such as sliders and buttons, it was not straightforward to understand for the end-user, especially when a series of strict steps must be followed to achieve a successful result. Finally, these tools cannot simulate the output kinetic behaviors rather than currently simple previews. In the future, I plan to add error reporting and warning mechanisms to the tool and provide instructional onboarding tutorials for the end-user to follow and learn about the user interface. Using a physics engine and library, I will implement interactive environments to simulate the resulting kinetic behaviors such as deformation and motion. Various intrinsic and extrinsic factors, such as applied external force, weight, and frictions, will also be considered and integrated into the simulation module.

7.2.3 *Application Limitations and Future Work*

In this dissertation, I demonstrated the potential of proposed spring-based mechanisms and tools through example applications that showcased how springs were deformed to support toys, tangible props, assistive tools, and custom I/O devices. These contribute to a broad research topic—fabricating kinetic objects for I/O. With a similar research approach, I will expand on my research work and contribute to a wider variety of personal fabrication applications beyond those only enabled by the spring-based mechanisms. In the future, I will explore novel design techniques and sort out software and hardware solutions (i) to

facilitate learning by combining AR/MR and actuated fabricated objects, (ii) to augment the communication and interaction between people and the physical environment, (iii) to create assistive mediums for people with impairments to interact with digital elements, (iv) to enhance the tangibility of virtual objects for immersive XR experience, and (v) to embed computation and intelligence into kinetic objects using machine learning.

Facilitating learning by combining MR and kinetic objects. Tangible, physical devices are great tools for people, especially children, to learn about new concepts and explore the surrounding world. For example, I developed a set of tangible props that helped children learn an old language by associating physical objects used in a forest in Malaysia with digital elements on a touchscreen [87] 7.1a. Another example is MakerWear [51], a modular tangible toolkit for children to create custom wearable devices 7.1b. I will continue exploring novel fabrication techniques and AR/MR technology to facilitate learning, especially when the fabricated objects are movable and reconfigurable using spring-based mechanisms or other kinetic structures.



Figure 7.1: Two past projects on interactive tangible interfaces for children to (a) learn about an old language and (b) build custom wearable devices.

Augmenting human-environment communication and interaction through dynamically controllable agents. One exciting application area is converting the surrounding environment into accessible space. I envision two ways: first, providing assistive

aids on the human body with fabricated devices. For example, we developed a pneumatically controlled armband that guided blind and low vision users to fetch target nearby objects via soft haptic clues on the skin [35, 37] 7.2. Second, retrofitting the existing environment with uniquely fabricated layers to augment human communication and interaction with the surroundings. For example, the ongoing project explores a novel digital fabrication pipeline that enables the user to design, review, and deploy in-situ 3D printable tactile surface patterns on demand. Our environments are filled with surfaces such as walls and curved poles, which can be utilized and augmented as potential interactive interfaces.

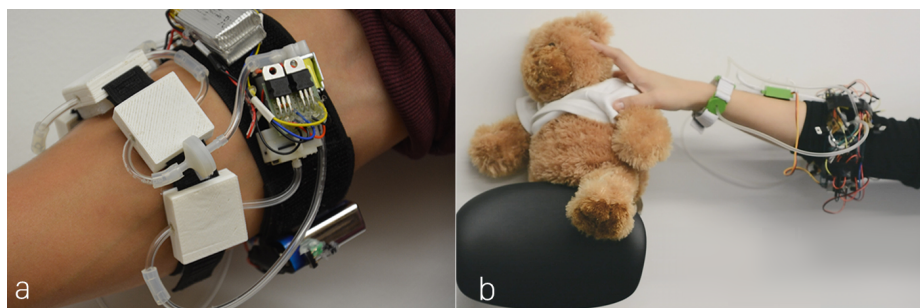


Figure 7.2: Two wearable devices that provide tactile feedback on body for blind and low vision users to (a) navigate and (b) fetch objects.

Creating assistive and dynamic mediums for accessibility. I believe that accessible and personalized interfaces and tools that leverage emerging kinetic fabrication techniques will boost the ways to access and manage pervasive yet constantly changing digital information. For example, I explored 3D-printed overlays for blind users to access graphical content on touchscreens and investigated how people perceive and react to digital content through 3D printing [34] 7.3a. With kinetic fabrication, I will develop kinetically-enabled assistive I/O devices to address real-world problems with a focus on accessibility issues and to facilitate user interactions with digital data on various platforms, such as leveraging a multi-model approach.

Enhancing tangibility in XR. As the boundaries between virtual and physical environments have blurred thanks to the rise of technologies for XR (*i.e.*, VR, AR, and MR), I plan to expand kinetic fabrication to enhance virtual objects' tangibility and support social

interactions in XR. For example, I explored how to provide dynamic tactile feedback on the forearm via a tangible puppet agent in a VR game [19] 7.3b and how to simulate accurate force feedback for various activities in a VR environment [65]. In this category, creating responsive and always-on interfaces to establish and facilitate interpersonal communications and collaborations in XR is still an open research area. For example, designers and engineers could co-design a 3D car model on a VR-based platform with the digital model updating and providing tangible feedback for all the users in real-time. Yet, another challenge is that, in a VR environment, the user cannot touch, feel, and physically manipulate virtual objects. With more custom on-demand devices enabled by kinetic fabrication techniques, I believe that users will have a more immersive experience in VR environments with both visual and tactile sensations.

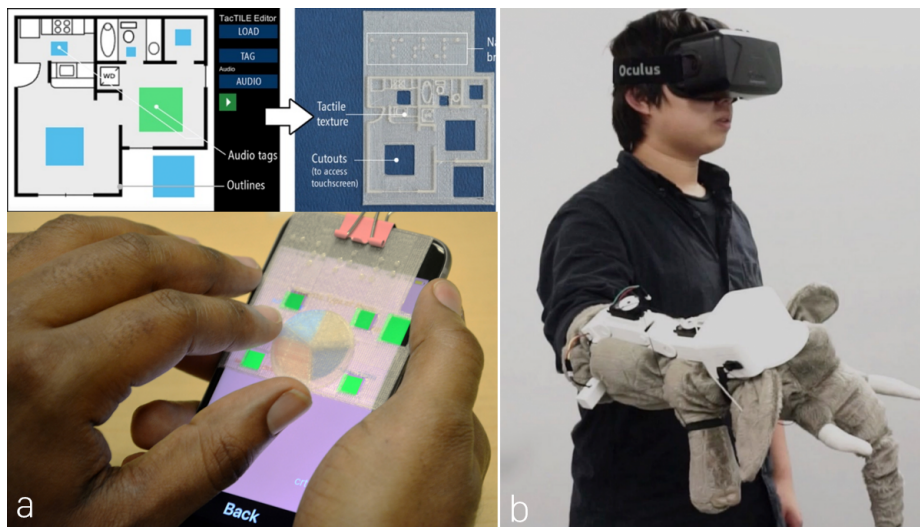


Figure 7.3: Two past projects that focused on (a) tactile overlays for making graphics accessible on touchscreens and (b) tactile wearable controller for a VR game.

Embedding computation and intelligence into kinetic objects using machine learning. The last direction I want to explore is to imbue machine learning into 3D-printed kinetic objects to create intelligent devices that interpret user input as commands, augmenting human-object interaction and communication. For example, in project SqueezaPulse [31], I used a single microphone to distinguish different air pulses generated by the user

squeezing soft and squeezable parts, which had silicone-casting cavities and were attached to 3D models. The air pulse traveled through flexible tubing, which connected the cavity and the microphone, and was then recognized as an interactive input by a pre-trained machine learning model 7.4a. In a recent project, I built a machine learning classifier to recognize different deformation behaviors of a 3D-printed metal helical spring, printed on the advanced Desktop Metal printer¹, through inductive sensing. The resulting metal helical spring was connected to a Texas Instruments LDC1614 evaluation board via two jumper wires and paralleled with a capacitor 7.4b. By feeding the received inductance signals into a machine learning model, I could classify six different spring deformation behaviors in addition to the default spring's equilibrium state: twisting, bending, short compressing, long compressing, soft compressing, and hard compressing. The trained shape-based classifier achieved an overall 91.43% accuracy. I believe that combining machine learning and kinetic objects will inspire new applications in the intersection of fabrication and sensing.

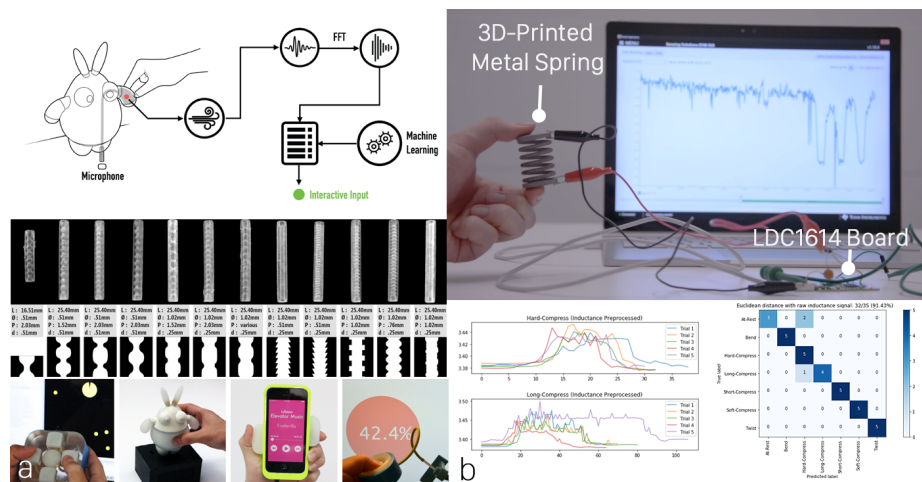


Figure 7.4: Two past projects that used machine learning to add interactivity to fabricated objects: (a) converting squeezing gestures into interactive user input using acoustic sensing and (b) classifying spring's deformation behaviors using inductive sensing.

¹Desktop Metal: <https://www.desktopmetal.com/>

7.3 Final Remarks

As 3D printing has become increasingly accessible and inexpensive, we have the opportunity to take advantage of the emerging personal fabrication technology to create demanding, personalized, and highly custom objects. However, 3D-printed objects are usually static, and it is difficult for casual users to create movable and functional kinetic 3D objects with 3D printing. This dissertation aims to offer a solution by creating a series of spring-based mechanisms to support the design, control, and fabrication of 3D printable kinetic objects for interaction. My hope is that the presented approaches in this dissertation will inspire more fabrication researchers, not only from the HCI community but also from other areas such as mechanical engineering, to pursue the research field of kinetic fabrication. I firmly believe that fabricating objects that can dynamically adapt to humans, objects, and the environment will be the future of personal fabrication. This dissertation is just the first step toward achieving such a bold vision.

BIBLIOGRAPHY

- [1] Celena Alcock, Nathaniel Hudson, and Parmit K. Chilana. Barriers to Using, Customizing, and Printing 3D Designs on Thingiverse. In *Proceedings of the 19th International Conference on Supporting Group Work*, GROUP '16, page 195–199, New York, NY, USA, 2016. Association for Computing Machinery.
- [2] Ashok G. Ambekar. Mechanism and Machine Theory. *Bohem Press*, 2007.
- [3] Davide Jose Nogueira Amorim, Troy Nachtigall, and Miguel Bruns Alonso. Exploring Mechanical Meta-material Structures Through Personalised Shoe Sole Design. *Proceedings: SCF 2019 - ACM Symposium on Computational Fabrication*, 2019.
- [4] Byoungkwon An, Ye Tao, Jianzhe Gu, Tingyu Cheng, Xiang 'Anthony' Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, Shigeo Takahashi, Hsiang-Yun Wu, Teng Zhang, and Lining Yao. Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Montreal QC Canada, April 2018. ACM.
- [5] Lisa Anthony, YooJin Kim, and Leah Findlater. Analyzing User-Generated Youtube Videos to Understand Touchscreen Use by People with Motor Impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 1223–1232, New York, NY, USA, 2013. Association for Computing Machinery.
- [6] Rafael Ballagas, Sarthak Ghosh, and James Landay. The Design Space of 3D Printable Interactivity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(2), jul 2018.
- [7] Patrick Baudisch and Stefanie Mueller. Personal Fabrication. *Foundations and Trends in Human-Computer Interaction*, 10(3-4):165–293, 2017.
- [8] Felix W. Baumann. Thirty Thousand 3D Models from Thingiverse. *Data*, 3(1), 2018.
- [9] D. Besnea, C. I. Rizescu, D. Rizescu, D. Comeaga, R. Ciobanu, and E. Moraru. Study of Deflection Behavior of 3D Printed Leaf Springs. *IOP Conference Series: Materials Science and Engineering*, 444(4), 2018.
- [10] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design and Fabrication of Materials with Desired Deformation Behavior. *ACM Transactions on Graphics*, 29(4):1–1, 2010.

- [11] Richard Budynas and J. Keith Nisbett. Shigley's Mechanical Engineering Design (8th Edition). *McGraw-Hill Science/Engineering/Math*, pages 499–548, 2008.
- [12] Erin Buehler, Stacy Branham, Abdullah Ali, Jeremy J. Chang, Megan Kelly Hofmann, Amy Hurst, and Shaun K. Kane. Sharing is Caring: Assistive Technology Designs on Thingiverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 525–534, New York, NY, USA, 2015. Association for Computing Machinery.
- [13] Moritz Bächer, Bernd Bickel, Doug L. James, and Hanspeter Pfister. Fabricating Articulated Characters from Skinned Meshes. *ACM Transactions on Graphics*, 31(4):1–9, 2012.
- [14] Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski. LinkEdit: Interactive Linkage Editing Using Symbolic Kinematics. *ACM Transactions on Graphics*, 34(4):1–8, 2015.
- [15] Moritz Bächer, Benjamin Hepp, Fabrizio Pece, Paul G. Kry, Bernd Bickel, Bernhard Thomaszewski, and Otmar Hilliges. DefSense: Computational Design of Customized Deformable Input Devices. *Conference on Human Factors in Computing Systems - Proceedings*, pages 3806–3816, 2016.
- [16] Jacques Calì, Dan A. Calian, and Rebecca Kleinberger. 3D-Printing of Non-Assembly, Articulated Models. *ACM Transactions on Graphics (TOG)*, 31(6):1–8, 2012.
- [17] Peter R.N. Childs. *Mechanical Design Engineering Handbook*. 2004. Pages: 159.
- [18] Ruta Desai, James McCann, and Stelian Coros. Assembly-Aware Design of Printable Electromechanical Devices. *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 457–472, 2018.
- [19] Ruofei Du and Liang He. Vrsurus: Enhancing interactivity and tangibility of puppets in virtual reality. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, page 2454–2461, New York, NY, USA, 2016. Association for Computing Machinery.
- [20] Jack Forman and Hamilton Forsythe. DefeXtiles: 3D Printing Quasi-Woven Fabric Via Under-Extrusion. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 1222–1233. Association for Computing Machinery, 2020.
- [21] Mark Fuge, Greg Carmean, Jessica Cornelius, and Ryan Elder. The MechProcessor: Helping Novices Design Printable Mechanisms Across Different Printers. *Journal of Mechanical Design*, 137(11):111415–111415, 2015.

- [22] Wei Gao, Yunbo Zhang, Diogo C. Nazzetta, Karthik Ramani, and Raymond J. Cipra. RevoMaker: Enabling Multi-Directional and Functionally-Embedded 3D Printing Using a Rotational Cuboidal Platform. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, pages 437–446, 2015.
- [23] Sharon Gerbode. Springs with Bones: Several New Discoveries in the Creation and Use of Nonlinear Springs. *Thesis. Unarchived.*, 2018.
- [24] Jun Gong, Olivia Seow, Cedric Honnet, Jack Forman, and Stefanie Mueller. MetaSense: Integrating Sensing Capabilities into Mechanical Metamaterial. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 1063–1073, Virtual Event USA, October 2021. ACM.
- [25] Daniel Groeger, Martin Feick, Anusha Withana, and Jürgen Steimle. Tactlets: Adding Tactile Feedback to 3D Objects Using Custom Printed Controls. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 923–936, 2019.
- [26] Daniel Groeger, Elena Chong Loo, and Jürgen Steimle. HotFlex: Post-Print Customization of 3D Prints Using Embedded State Change. *Conference on Human Factors in Computing Systems - Proceedings*, pages 420–432, 2016.
- [27] Jianzhe Gu, David E Breen, Jenny Hu, Lifeng Zhu, Ye Tao, Tyson Van De Zande, and Guanyun Wang. Geodesy: Self-Rising 2.5D Tiles by Printing Along 2D Geodesic Closed Path. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2019.
- [28] Jianzhe Gu, Yuyu Lin, Qiang Cui, Xiaoqian Li, Jiaji Li, Lingyun Sun, Cheng Yao, Fangtian Ying, Guanyun Wang, and Lining Yao. PneuMesh: Pneumatic-Driven Truss-Based Shape Changing System. In *CHI Conference on Human Factors in Computing Systems*, pages 1–12, New Orleans LA USA, April 2022. ACM.
- [29] Jianzhe Gu, Vidya Narayanan, Guanyun Wang, Danli Luo, Harshika Jain, Kexin Lu, Fang Qin, Sijia Wang, James McCann, and Lining Yao. Inverse Design Tool for Asymmetrical Self-Rising Surfaces with Color Texture. pages 1–12, 2020.
- [30] Liang He, Jessica Chin, and Jon E. Froehlich. Making Things Move: Analyzing How Makers Create 3D-Printable Kinetic Designs on Thingiverse. *Planned for Submission to CHI 2023*, 2022.
- [31] Liang He, Gierad Laput, Eric Brockmeyer, and Jon E. Froehlich. SqueezaPulse: Adding Interactive Input to Fabricated Objects Using Corrugated Tubes and Air Pulses. *Proceedings of the 11th ACM International Conference on Tangible, Embedded, and Embodied Interaction, TEI 2017*, pages 341–350, 2017.

- [32] Liang He, Huaishu Peng, Michelle Lin, Ravikanth Konjeti, François Guimbretière, and Jon E. Froehlich. Ondulé: Designing and Controlling 3D Printable Springs. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 739–750, New Orleans LA USA, October 2019. ACM.
- [33] Liang He, Xia Su, Huaishu Peng, Jeffrey I. Lipton, and Jon E. Froehlich. Kinergy: Creating 3D Printable Motion using Embedded Kinetic Energy. *To appear at the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29–November 2, 2022, Bend, OR, USA, 2022.*
- [34] Liang He, Zijian Wan, Leah Findlater, and Jon E. Froehlich. TacTILE: A Preliminary Toolchain for Creating Accessible Graphics with 3D-Printed Overlays and Auditory Annotations. *ASSETS 2017 - Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 397–398, 2017.
- [35] Liang He, Ruolin Wang, and Xuhai Xu. PneuFetch: Supporting Blind and Visually Impaired People to Fetch Nearby Objects via Light Haptic Cues. *Conference on Human Factors in Computing Systems - Proceedings*, pages 1–9, 2020.
- [36] Liang He, Jarrid A. Wittkopf, Ji Won Jun, Kris Erickson, and Rafael Tico Ballagas. ModElec: A Design Tool for Prototyping Physical Computing Devices Using Conductive 3D Printing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(4), dec 2022.
- [37] Liang He, Cheng Xu, Ding Xu, and Ryan Brill. PneuHaptic: Delivering Haptic Cues with a Pneumatic Armband. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers, ISWC '15*, page 47–48, New York, NY, USA, 2015. Association for Computing Machinery.
- [38] Jean Hergel and Sylvain Lefebvre. 3D Fabrication of 2D Mechanisms. *Computer Graphics Forum*, 34(2):229–238, 2015.
- [39] Freddie Hong, Connor Myant, and David Boyle. Thermoformed Circuit Boards: Fabrication of Highly Conductive Freeform 3D Printed Circuit Boards with Heat Bending. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–10, May 2021. arXiv:2011.06473 [cs].
- [40] Scott E Hudson. Printing Teddy Bears : A Technique for 3D Printing of Soft Interactive Objects. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 459–468, 2014.
- [41] Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang Ting Chen, and Patrick Baudisch. Metamaterial Mechanisms. *UIST 2016 - Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 529–539, 2016.

- [42] Alexandra Ion, Robert Kovacs, Oliver S. Schneider, Pedro Lopes, and Patrick Baudisch. Metamaterial Textures. *Conference on Human Factors in Computing Systems - Proceedings*, 2018-April:1–12, 2018.
- [43] Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. Understanding Metamaterial Mechanisms. *Conference on Human Factors in Computing Systems - Proceedings*, 2019.
- [44] Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. Digital Mechanical Metamaterials. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, pages 977–988, 2017.
- [45] Miyu Iwafune, Taisuke Ohshima, and Yoichi Ochiai. Coded Skeleton : Programmable Deformation Behaviour for Shape Changing Interfaces. pages 2–3, 2016.
- [46] Vikram Iyer, Justin Chan, Ian Culhane, Jennifer Mankoff, and Shyamnath Gollakota. Wireless Analytics for 3D Printed Objects. *UIST 2018 - Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 141–152, 2018.
- [47] Vikram Iyer, Justin Chan, and Shyamnath Gollakota. 3D Printing Wireless Connected Objects. *ACM Transactions on Graphics*, 36(6), 2017.
- [48] Faizan Javed, Arslan Khalid, and Atique Ahmad. Design and Analysis of a 3D Printed Compliant Modified Jansen’s Linkage by Using Cartwheel Flexures. *2020 International Symposium on Recent Advances in Electrical Engineering Computer Sciences (RAEE CS)*, 5:1–5, 2020.
- [49] Michael D. Jones, Zann Anderson, Casey Walker, and Kevin Seppi. PHUI-Kit: Interface Layout and Fabrication on Curved 3D Printed Objects. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018-April:1–11, 2018.
- [50] Hiroki Kaimoto, Junichi Yamaoka, Satoshi Nakamaru, Yoshihiro Kawahara, and Yasuaki Kakehi. ExpandFab: Fabricating Objects Expanding and Changing Shape with Heat. *Conference on Tangible, Embedded, and Embodied Interaction*, pages 153–164, 2020.
- [51] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E. Froehlich. MakerWear: A Tangible Approach to Interactive Wearable Creation for Children. *Conference on Human Factors in Computing Systems - Proceedings*, 2017-May:133–145, 2017.
- [52] Rubaiat Habib Kazi, Tovi Grossman, Cory Mogk, Ryan Schmidt, and George Fitzmaurice. ChronoFab: Fabricating Motion. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 908–918, 2016.

- [53] Jeeun Kim, Qingnan Zhou, Amanda Ghassaei, and Xiang 'Anthony' Chen. OmniSoft: A Design Tool for Soft Objects by Example. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 1–13, Salzburg Austria, February 2021. ACM.
- [54] Donghyeon Ko, Jee Bin Yim, Yujin Lee, Jaehoon Pyun, and Woohun Lee. Designing Metamaterial Cells to Enrich Thermoforming 3D Printed Object for Post-Print Modification. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Yokohama Japan, May 2021. ACM.
- [55] Robert Kovacs, Alexandra Ion, Pedro Lopes, Tim Oesterreich, Johannes Filter, Philip Otto, Tobias Arndt, Nico Ring, Melvin Witte, Anton Synytsia, and Patrick Baudisch. TrussFormer: 3D Printing Large Kinetic Structures. *UIST 2018 - Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 113–125, 2018.
- [56] Robert Kovacs, Lukas Rambold, Lukas Fritzsche, Dominik Meier, Jotaro Shigeyama, Shohei Katakura, Ran Zhang, and Patrick Baudisch. Trusscillator: A System for Fabricating Human-Scale Human-Powered Oscillating Devices. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 1074–1088, Virtual Event USA, October 2021. ACM.
- [57] Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläslius, and Patrick Baudisch. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. *CHI '17 Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, (Figure 1):2606–2616, 2017.
- [58] Gierad Laput, Eric Brockmeyer, Moshe Mahler, Scott E. Hudson, and Chris Harrison. Acoustruments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.
- [59] Danny Leen, Nadya Peek, and Raf Ramakers. *LamiFold: Fabricating Objects with Integrated Mechanisms Using a Laser Cutter Lamination Workflow*, page 304–316. Association for Computing Machinery, New York, NY, USA, 2020.
- [60] Dingzeyu Li, David I.W. Levin, Wojciech Matusik, and Changxi Zheng. Acoustic Voxels: Computational Optimization of Modular Acoustic Filters. *ACM Transactions on Graphics*, 35(4), 2016.
- [61] Jiahao Li, Meilin Cui, Jeeun Kim, and Xiang Anthony Chen. Romeo: A Design Tool for Embedding Transformable Parts in 3D Models to Robotically Augment Default Functionality. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 897–911, 2020.

- [62] Jiahao Li, Jeeun Kim, and Xiang 'Anthony' Chen. Robiot: A Design Tool for Actuating Everyday Objects with Automatically Generated 3D Printable Mechanisms. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 673–685, 2019.
- [63] Jiahao Li, Alexis Samoylov, Jeeun Kim, and Xiang 'Anthony' Chen. Roman: Making Everyday Objects Robotically Manipulable with 3D-Printable Add-on Mechanisms. In *CHI Conference on Human Factors in Computing Systems*, pages 1–17, New Orleans LA USA, April 2022. ACM.
- [64] Nianlong Li, Han Jong Kim, Luyao Shen, Feng Tian, Teng Han, Xing Dong Yang, and Tek Jin Nam. HapLinkage: Prototyping Haptic Proxies for Virtual Hand Tools Using Linkage Mechanism. *UIST 2020 - Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 1261–1274, 2020.
- [65] Hongnan Lin, Liang He, Fangli Song, Yifan Li, Tingyu Cheng, Clement Zheng, Wei Wang, and HyunJoo Oh. FlexHaptics: A Design Method for Passive Haptic Inputs Using Planar Compliant Structures. In *CHI Conference on Human Factors in Computing Systems*, pages 1–13, New Orleans LA USA, April 2022. ACM.
- [66] Min Liu, Yunbo Zhang, Jing Bai, Yuanzhi Cao, Jeffrey M. Alperovich, and Karthik Ramani. WireFab: Mix-Dimensional Modeling and Fabrication for 3D Mesh Models. *Conference on Human Factors in Computing Systems*, pages 965–976, 2017.
- [67] Li Ke Ma, Yizhong Zhang, Yang Liu, Kun Zhou, and Xin Tong. Computational Design and Fabrication of Pneumatic Objects with Desired Deformations. *ACM Transactions on Graphics*, 36(6), 2017.
- [68] Robert Maccurdy, Robert Katzschmann, Youbin Kim, and Daniela Rus. Printable Hydraulics: A Method for Fabricating Robots by 3D Co-Printing Solids and Liquids. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June:3878–3885, 2016.
- [69] Eric MacDonald and Ryan Wicker. Multiprocess 3D Printing for Increasing Component Functionality. *Science*, 353(6307):aaf2093, 2016.
- [70] Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. FlexMaps: Computational Design of Flat Flexible Shells for Shaping 3D Objects. *SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia 2018*, 37(6), 2018.
- [71] Vittorio Megaro, Bernhard Thomaszewski, Maurizio Nitti, Otmar Hilliges, Markus Gross, and Stelian Coros. Interactive Design of 3D-Printable Robotic Creatures. *ACM Transactions on Graphics*, 34(6):1–9, 2015.

- [72] Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus Gross, and Bernhard Thomaszewski. A Computational Design Tool for Compliant Mechanisms. *ACM Transactions on Graphics*, 36(4), 2017.
- [73] Nicholas A. Meisel, Amelia M. Elliott, and Christopher B. Williams. A Procedure for Creating Actuated Joints via Embedding Shape Memory Alloys in PolyJet 3D Printing. *Journal of Intelligent Material Systems and Structures*, 26(12):1498–1512, 2015.
- [74] Devon J. Merrill, Jorge Garza, and Steven Swanson. Echidna: Mixed-Domain Computational Implementation via Decision Trees. *Proceedings: SCF 2019 - ACM Symposium on Computational Fabrication*, 2019.
- [75] Niloy J. Mitra, Yong Liang Yang, Dong Ming Yan, Wilmot Li, and Maneesh Agrawala. Illustrating How Mechanical Assemblies Work. *ACM SIGGRAPH 2010 Papers, SIGGRAPH 2010*, 3, 2010.
- [76] Lucio S. Nahum. Off-Line Sensing: Memorizing Interactions in Passive 3D-Printed Objects. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–8, 2018.
- [77] Ken Nakagaki, Joanne Leong, Jordan L Tappa, João Wilbert, and Hiroshi Ishii. HERMITS: Dynamically Reconfiguring the Interactivity of Self-Propelled TUIs with Mechanical Shell Add-Ons. *Proceedings of Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 882–896, 2020.
- [78] Ryuma Niiyama, Hiroki Sato, Kazumasa Tsujimura, Koya Narumi, Young Ah Seong, Ryosuke Yamamura, Yasuaki Kakehi, and Yoshihiro Kawahara. Poimo: Portable and Inflatable Mobility Devices Customizable for Personal Physical Characteristics. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020.
- [79] Yuta Noma, Koya Narumi, Fuminori Okuya, and Yoshihiro Kawahara. Pop-up Print: Rapidly 3D Printing Mechanically Reversible Objects in the Folded State. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 58–70, 2020.
- [80] Hyunjoon Oh, Tung D. Ta, Ryo Suzuki, Mark D. Gross, Yoshihiro Kawahara, and Lining Yao. PEP (3D Printed Electronic Papercrafts): An Integrated Approach for 3D Sculpting Paper-Based Electronic Devices. pages 1–12, 2018.
- [81] Ahmed Othman, Steven Hartman, Dragan Ströbele, Jassin Arnold, and Constantin von See. Preliminary Feasibility Torque Mechanical Evaluation for 3D Printed Orthodontic Springs with Different Parameters: Invitro Study. 2020.

- [82] Jifei Ou, Gershon Dublon, Chin-yi Cheng, Felix Heibeck, Karl Willis, and Hiroshi Ishii. Cillia: 3D Printed Micro-Pillar Structures for Surface Texture, Actuation and Sensing. *Conference on Human Factors in Computing Systems*, pages 5753–5764, 2016.
- [83] Jifei Ou, Zhao Ma, Jannik Peters, Sen Dai, Nikolaos Vlavianos, and Hiroshi Ishii. KinetiX: Designing Auxetic-Inspired Deformable Material Structures. *Computers and Graphics (Pergamon)*, 75:72–81, 2018. Publisher: Elsevier Ltd.
- [84] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Dennis Zorin. Elastic Textures for Additive Fabrication. *ACM Transactions on Graphics*, 34(4):135:1–135:12, 2015.
- [85] Huaishu Peng, François Guimbretière, James McCann, and Scott E. Hudson. A 3D Printer for Interactive Electromagnetic Devices. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 553–562, 2016.
- [86] Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. A Layered Fabric 3D Printer for Soft Interactive Objects. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1789–1798, 2015.
- [87] Beryl Plimmer, Liang He, Tariq Zaman, Kasun Karunanayaka, Alvin W. Yeo, Garen Jengan, Rachel Blagojevic, and Ellen Yi-Luen Do. New Interaction Tools for Preserving an Old Language. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, page 3493–3502, New York, NY, USA, 2015. Association for Computing Machinery.
- [88] Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. RetroFab: A Design Tool for Retrofitting Physical Interfaces Using Actuators, Sensors and 3D Printing. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 409–419, 2016.
- [89] Dan Raviv, Wei Zhao, Carrie McKnelly, Athina Papadopoulou, Achuta Kadambi, Boxin Shi, Shai Hirsch, Daniel Dikovsky, Michael Zyracki, Carlos Olguin, Ramesh Raskar, and Skylar Tibbits. Active Printed Materials for Complex Self-Evolving Deformations. *Scientific Reports*, 4:1–8, 2014.
- [90] Michael L. Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E. Hudson. Stretching the Bounds of 3D Printing with Embedded Textiles. *Conference on Human Factors in Computing Systems - Proceedings*, 2017-May:497–508, 2017.
- [91] Claudia Daudén Roquet, Jeeun Kim, and Tom Yeh. 3D Folded PrintGami: Transform Passive 3D Printed Objects to Interactive by Inserted Paper Origami Circuits.

- Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 187–191, 2016.
- [92] Thijs Jan Roumen, Willi Müller, and Patrick Baudisch. Grafter: Remixing 3D-Printed Machines. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, (c):1–12, 2018.
- [93] Enea Sacco. Analysis and Modelling of 3D Printed Springs for Use in Spacecraft. *PhD diss.*, 2020.
- [94] Andrew O. Sageman-Furnas, Nobuyuki Umetani, and Ryan Schmidt. Meltables: Fabrication of Complex 3D Curves by Melting. *SIGGRAPH Asia 2015 Technical Briefs, SA 2015*, pages 1–4, 2015.
- [95] Harpreet Sareen, Udayan Umapathi, Patrick Shin, Yasuaki Kakehi, Jifei Ou, Hiroshi Ishii, and Pattie Maes. Printflatables: Printing Human-Scale, Functional and Dynamic Inflatable Objects. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, pages 3669–3680, 2017.
- [96] Valkyrie Savage, Colin Chang, and Björn Hartmann. Sauron: Embedded Single-Camera Sensing of Printed Physical User Interfaces. *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 447–456, 2013.
- [97] Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. Makers' Marks: Physical Markup for Designing and Fabricating Functional Objects. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, pages 103–108, 2015.
- [98] Valkyrie Savage, Andrew Heady, Björn Hartmann, Dan B. Goldman, Gautham Mysore, and Wilmot Li. Lamello: Passive Acoustic Sensing for Tangible Input Components. *Conference on Human Factors in Computing Systems - Proceedings*, 2015-April:1277–1280, 2015.
- [99] Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. A Series of Tubes: Adding Interactivity to 3D Prints Using Internal Pipes. *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 3–12, 2014.
- [100] Umberto Scarcia, Manuele Ghinelli, Claudio Melchiorri, Giovanni Berselli, and Gianluca Palli. Optimal Design of 3D Printed Spiral Torsion Springs. *Proceedings of the ASME 2016 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, 50497:V002T03A020–V002T03A020, 2016.

- [101] Martin Schmitz, Urgen Steimle, Jochen Huber, Niloofar Dezfuli, and Max Uhlhäuser. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1001–1014, 2017.
- [102] Martin Schmitz, Martin Stitz, Florian Müller, Markus Funk, and Max Mühlhäuser. ./Trilaterate: A Fabrication Pipeline to Design and 3D Print Hover-, Touch-, and Force-Sensitive Objects. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.
- [103] Adriana Schulz, Jie Xu, Bo Zhu, Changxi Zheng, Eitan Grinspun, and Wojciech Matusik. Interactive Design Space Exploration and Optimization for CAD Models. *ACM Transactions on Graphics*, 36(4), 2017.
- [104] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. Microstructures to Control Elasticity in 3D Printing. *ACM Transactions on Graphics*, 34(4), 2015.
- [105] Christian Schumacher, Steve Marschner, Markus Gross, and Bernhard Thomaszewski. Mechanical Characterization of Structured Sheet Materials. *ACM Transactions on Graphics*, 37(4):1–15, 2018.
- [106] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. Computational Design of Actuated Deformable Characters. *ACM Transactions on Graphics*, 32(4):1–10, 2013.
- [107] Ronit Slyper and Jessica Hodgins. Prototyping Robot Appearance, Movement, and Interactions Using Flexible 3D Printing and Air Pressure Sensors. *2012 IEEE RO-MAN: The 21st IEEE International Workshop on Robot and Human Interactive Communication*, 1:6–11, 2012.
- [108] Peng Song, Xiaofei Wang, Xiao Tang, Chi-Wing Fu, Hongfei Xu, Ligang Liu, and Niloy J. Mitra. Computational Design of Wind-up Toys. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.
- [109] Hiroshi Sugihara. Ready to Crawl. 2016.
- [110] Lingyun Sun, Jiaji Li, Yu Chen, Yue Yang, Zhi Yu, Danli Luo, Jianzhe Gu, Lining Yao, Ye Tao, and Guanyun Wang. FlexTruss: A Computational Threading Method for Multi-material, Multi-form and Multi-use Prototyping. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Yokohama Japan, May 2021. ACM.

- [111] Saiganesh Swaminathan, Kadri Bugra Ozutemiz, Carmel Majidi, and Scott E. Hudson. FiberWire: Embedding Electronic Function into 3D Printed Mechanically Strong, Lightweight Carbon Fiber Composite Objects. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, (Figure 1):1–11, 2019.
- [112] Haruki Takahashi and Jeeun Kim. 3D Printed Fabric: Techniques for Design and 3D Weaving Programmable Textiles. *UIST 2019 - Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 43–51, 2019.
- [113] Kazuki Takazawa, Satoshi Hashizume, Ryuichiro Sasaki, Yoshikuni Hashimoto, and Yoichi Ochiai. Morpho Sculptures: Digital Fabrication Methods of Engraving Flat Materials into Shape Changing User Interfaces. *ACM SIGGRAPH 2017 Posters, SIGGRAPH 2017*, pages 3–4, 2017.
- [114] Kengo Tanaka, Taisuke Ohshima, and Yoichi Ochiai. Spring-Pen: Reproduction of Any Softness with the 3D Printed Spring. *SIGGRAPH Asia 2017 Posters, SA 2017*, pages 3–4, 2017.
- [115] Ye Tao, Youngwook Do, Humphrey Yang, Yi Chin Lee, Guanyun Wang, Catherine Mondoa, Jianxun Cui, Wen Wang, and Lining Yao. Morphlour: Personalized Flour-Based Morphing Food Induced by Dehydration or Hydration Method. *UIST 2019 - Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 329–340, 2019.
- [116] Ye Tao, Jianzhe Gu, Byoungkwon An, Tingyu Cheng, Xiang Anthony Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, Teng Zhang, and Lining Yao. Demonstrating Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. *Conference on Human Factors in Computing Systems - Proceedings*, 2018-April, 2018.
- [117] Md. Farhan Tasnim Oshim, Julian Killingback, Dave Follette, Huaishu Peng, and Tauhidur Rahman. MechanoBeat: Monitoring Interactions with Everyday Objects Using 3D Printed Harmonic Oscillators and Ultra-Wideband Radar. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 430–444, 2020.
- [118] Carlos E. Tejada, Raf Ramakers, Sebastian Boring, and Daniel Ashbrook. AirTouch: 3D-Printed Touch-Sensitive Objects Using Pneumatic Sensing. *Conference on Human Factors in Computing Systems - Proceedings*, 2020.
- [119] Nobuyuki Umetani, Athina Panotopoulou, Ryan Schmidt, and Emily Whiting. Print-one: Interactive Resonance Simulation for Free-Form Print-Wind Instrument Design. *ACM Transactions on Graphics (TOG)*, 35(6):1–14, 2016.
- [120] Nobuyuki Umetani and Ryan Schmidt. SurfCuit: Surface-Mounted Circuits on 3D Prints. *IEEE Computer Graphics and Applications*, 38(3):52–60, 2017.

- [121] Francisca Gil Ureta, Chelsea Tymms, and Denis Zorin. Interactive Modeling of Mechanical Objects. *Eurographics Symposium on Geometry Processing*, 35(5):145–155, 2016.
- [122] Marynel Vázquez, Eric Brockmeyer, Ruta Desai, Chris Harrison, and Scott E. Hudson. 3D Printing Pneumatic Device Controls with Variable Activation Force Capabilities. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pages 1295–1304, 2015.
- [123] Guanyun Wang, Fang Qin, Haolin Liu, Ye Tao, Yang Zhang, Yongjie Jessica Zhang, and Lining Yao. Morphingcircuit: An Integrated Design, Simulation, and Fabrication Workflow for Self-Morphing Electronics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4), 2020.
- [124] Guanyun Wang, Ye Tao, Ozguc Bertug Capunaman, Humphrey Yang, and Lining Yao. A-line: 4D Printing Morphing Linear Composite Structures. *Conference on Human Factors in Computing Systems*, pages 1–12, 2019.
- [125] Guanyun Wang, Humphrey Yang, Zeyu Yan, Nurcan Gecer Ulu, Ye Tao, Jianzhe Gu, Levent Burak Kara, and Lining Yao. 4DMesh: 4D Printing Morphing Non-Developable Mesh Surfaces. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 623–635, Berlin Germany, October 2018. ACM.
- [126] Karl D.D. Willis, Eric Brockmeyer, Scott E. Hudson, and Ivan Poupyrev. Printed Optics: 3D Printing of Embedded Optical Elements for Interactive Devices. *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, pages 589–598, 2012.
- [127] Chang Xiao, Karl Bayer, Changxi Zheng, and Shree K. Nayar. Vidgets: Modular Mechanical Widgets for Mobile Devices. *ACM Transactions on Graphics*, 38(4), 2019.
- [128] Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. Bend-it: Design and Fabrication of Kinetic Wire Characters. *ACM Transactions on Graphics*, 37(6):1–15, 2018.
- [129] Mingliang Xu, Mingyuan Li, Weiwei Xu, Zhigang Deng, Yin Yang, and Kun Zhou. Interactive Mechanism Modeling from Multi-View Images. *ACM Transactions on Graphics*, 35(6):1–13, November 2016.
- [130] Junichi Yamaoka, Mustafa Doga Dogan, Katarina Bulovic, Kazuya Saito, Yoshihiro Kawahara, Yasuaki Kakehi, and Stefanie Mueller. FoldTronics: Creating 3D Objects with Integrated Electronics Using Foldable Honeycomb Structures. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

- [131] Zeyu Yan and Huaishu Peng. FabHydro: Printing Interactive Hydraulic Devices with an Affordable SLA 3D Printer. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 298–311, Virtual Event USA, October 2021. ACM.
- [132] Humphrey Yang, Tate Johnson, Ke Zhong, Dinesh Patel, Gina Olson, Carmel Majidi, Mohammad Islam, and Lining Yao. ReCompFig: Designing Dynamically Reconfigurable Kinematic Devices Using Compliant Mechanisms and Tensioning Cables. In *CHI Conference on Human Factors in Computing Systems*, pages 1–14, New Orleans LA USA, April 2022. ACM.
- [133] Lining Yao, Ryuma Niyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. PneuUI: Pneumatically Actuated Soft Composite Materials for Shape Changing Interfaces. *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*, pages 13–22, 2013.
- [134] Christopher Yu, Keenan Crane, and Stelian Coros. Computational Design of Telescoping Structures. *ACM Transactions on Graphics*, 36(4):83:1–83:9, 2017.
- [135] Minjing Yu, Zipeng Ye, Yong-Jin Liu, Ying He, and Charlie CL Wang. LineUp: Computing Chain-Based Physical Transformation. *ACM Transactions on Graphics (TOG)*, 38(1):1–16, 2019.
- [136] Ye Yuan, Changxi Zheng, and Stelian Coros. Computational Design of Transformables. *Computer Graphics Forum*, 37(8):103–113, 2018.
- [137] Jonas Zehnder, Espen Knoop, Moritz Bächer, and Bernhard Thomaszewski. MetaSilicone: Design and Fabrication of Composite Silicone with Desired Mechanical Properties. *ACM Transactions on Graphics*, 36(6), 2017.
- [138] Ran Zhang, Thomas Auzinger, Duygu Ceylan, Wilmot Li, and Bernd Bickel. Functionality-Aware Retargeting of Mechanisms to 3D Shapes. *ACM Transactions on Graphics*, 36(4):1–13, 2017.
- [139] Yang Zhang, Gierad Laput, and Chris Harrison. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. *Conference on Human Factors in Computing Systems - Proceedings*, 2017-May:1–14, 2017.
- [140] Clement Zheng, Zhen Zhou Yong, Hongnan Lin, HyunJoo Oh, and Ching Chiuan Yen. Shape-Haptics: Planar & Passive Force Feedback Mechanisms for Physical Interfaces. In *CHI Conference on Human Factors in Computing Systems*, pages 1–15, New Orleans LA USA, April 2022. ACM.
- [141] Qingnan Zhou and Alec Jacobson. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797*, 2016.

- [142] Yahan Zhou, Shinjiro Sueda, Wojciech Matusik, and Ariel Shamir. Boxelization: Folding 3D Objects into Boxes. *ACM Transactions on Graphics*, 33(4):1–8, 2014.
- [143] Junyi Zhu, Lotta-Gili Blumberg, Yunyi Zhu, Martin Nisser, Ethan Levi Carlson, Xin Wen, Kevin Shum, Jessica Ayeley Quaye, and Stefanie Mueller. CurveBoards: Integrating Breadboards into Physical Objects to Prototype Function in the Context of Form. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.
- [144] Junyi Zhu, Yunyi Zhu, Jiaming Cui, Leon Cheng, Jackson Snowden, Mark Choumlakone, Michael Wessely, and Stefanie Mueller. MorphSensor: A 3D Electronic Design Tool for Reforming Sensor Modules. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 541–553, 2020.
- [145] Lifeng Zhu, Weiwei Xu, John Snyder, Yang Liu, Guoping Wang, and Baining Guo. Motion-Guided Mechanical Toy Modeling. *ACM Transactions on Graphics*, 31(6):1–10, 2012.

Appendix A

THE QUALITATIVE CODEBOOK AND ANALYSIS SOURCE

A.1 Category 1: Model Category/Purpose

Thingiverse Category: A Thingiverse-defined, mutually exclusive category code selected by makers upon uploading their model. This was collected from the *Thingiverse field*. The codes include "Art", "Fashion", "Gadgets", "Hobby", "Household", "Learning", "Models", "Tools", "Toys Games", "3D Printing", "Other".

Making Purpose: A qualitative code applied by our team based on how the maker describes the purpose of their design. This was collected from the *Summary field*. The codes include "sculpture", "re-make", "gift", "accessory", "decoration", "fabric", "experiment", "attachment", "others".

A.2 Category 2: Kinetic Component Design

Mechanism: Type of 3D-printable kinetic mechanism. This was collected from *3D model files*, the *Summary field*, *uploaded images*, and *videos*. The codes include "hinge", "slider", "joint", "gear", "telescoping structure", "bearing", "spring", "linkage", "crank", "interlocking structure", "lever", "microstructure", "cam", "slider", "axel", "others".

Material: Type of 3D-printable material. This was collected from the *Summary field* and *videos*. The codes include "PLA", "ABS", "PETG", "TPE", "NinjaFlex", "TPU", etc.

External Hardware: Type of external hardware components (if any). This was collected from the *Summary field*, *uploaded images*, and *videos*. The codes include "screw", "rod", "bearing", "spring", "electronics", "band", "string", "fabric", etc.

Actuator: How is the kinetic mechanism activated? This was collected from the *Summary field*, *uploaded images*, and *videos*. The codes include "human power", "electromotor", "weight", "air pressure", "spring", etc.

A.3 Category 3: Kinetic Behavior

Deformation: Type of deformation behavior (if any). This was collected from the *Summary field*, *uploaded images*, and *videos*. The codes include "bend", "twist", "articulate", "fold", "stretch/extend", "compress/squeeze", "transform".

Motion: Type of motion behavior (if any). This was collected from the *Summary field*, *uploaded images*, and *videos*. The codes include "translate", "rotate", "oscillate/reciprocate".

A.4 Category 4: Design Fabrication

Design Tool: The design tool used to create the model. This was collected from the *Summary field* and the *Print Settings field*. The codes include *Fusion360*, *Rhino*, *Solidworks*, etc.

3D Printer: The 3D printer used by the original uploader. This was collected from the *Summary field* and the *print settings field*. The codes include "Prusa", "MakerBot", "Creality Ender", etc.

Resolution: Print resolution used by the original uploader. This was collected from the *Summary field* and the *Print Settings field*. The codes include "0.1mm", "0.15mm", "0.2mm", etc.

Infill Density: Infill density used by the original uploader. This was collected from the *Summary field* and the *Print Settings field*. The codes include "solid", "10%", "20%", etc.

Supports: Whether supports are needed for the print. This was collected from the *Summary field* and the *uploaded 3D models*. The codes include "yes", "no", "doesn't matter".

Post-Print Process: Whether the model requires post-processing. This was collected from the *Summary field*, *uploaded images*, and *videos*. The codes include "assembly", "sanding/filing", "cutting", "lubricating", "gluing", "circuit integration", "no post-processing", "others".

A.5 Category 5: Social Interaction

Likes: Number of acquired 'likes' on the model. This was collected from the *Thingiverse field*. The code was the *numerical count*.

Makes: Number of users who clicked on the 'I made this' button. This was collected from the *Thingiverse field*. The code was the *numerical count*.

Collects: Number of collections this model was added to. This was collected from the *Thingiverse field*. The code was the *numerical count*.

Remixes: Number of times this model was 'remixed'. This was collected from the *Thingiverse field*. The code was the *numerical count*.

Popularity: The sum of 'likes', 'collects', 'makes', and 'remixes'. This was derived from the above four. The code was the *numerical count*.

Appendix B

HELICAL SPRING THEORY AND MECHANICAL EXPERIMENTS FOR Ondulé

B.1 Terminology and Concepts in Helical Spring Theory

Young's modulus (E) is a measure of an object's resistance to being deformed elastically (*i.e.*, non-permanently) with applied stress. E is defined as the ratio of tensile stress σ (the stress state leading to expansion) to tensile strain ϵ (the relative length of deformation under tensile force)—see Figure B.1a and Eq. 3.

Shear modulus (G) measures an object's tendency to shear when acted upon by opposing forces. G is defined as the ratio of *shear stress* (the stress state leading to shear parallel to the cross-section of the material) to *shear strain* (the relative length of deformation under shear force)—see Figure B.1b and Eq. 4.

Poisson ratio (ν) measures how much a material expands perpendicular to the direction of compression or extension. The relationship between E and G can be derived using ν —see Figure B.1c and Eq. 5.

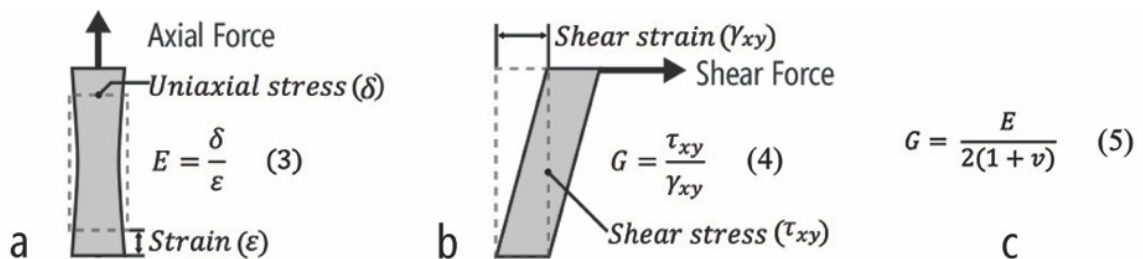


Figure B.1: Material properties (a) Young's modulus E and (b) shear modulus G . G can be derived using E and Poisson ratio (c).

B.2 Mechanical Experiment 1 Setup and Results

Following ASTM guidelines, I printed 60 solid test rods (Figure 4.3). I created two specimens for each combination and varied infill density, infill pattern, and printing orientation. Although varying infill patterns do not significantly impact material properties, I included this condition for completeness and compared line *vs.* triangle infills with 20% densities.

For the experiment itself, we used an *Instron 5585H 250kN* electro-mechanical load frame (Figure 4.3a), which works by gripping a test specimen and performing computer-controlled mechanical operations like stretching and compressing. In our case, we loaded individual test rods and performed a controlled tensile (stretching) operation, which separated the grips at 30mm/minute. The load frame’s data logger recorded the applied load, grip displacement, tensile stress σ , and tensile strain ϵ at 10Hz. Using these measurements, the load frame automatically calculates E (from Eq. 3). To measure ν , each test rod was also instrumented with two additional digital sensors: an *Instron 2630-106* axial extensometer and an *Instron 2640-008* transverse extensometer. Under tensile stress, the axial extensometer measured changes in rod length (axial elongation), and the transverse extensometer measured changes in rod diameter (transverse elongation). Both were also logged at 10Hz.

For infill density, we expect that as density increases, Young’s modulus E and the shear modulus G will also increase. That is, as the 3D-printed object becomes more solid, the force required to stretch or shear increases. Indeed, this is what I found: Figure 4.5 shows that regardless of printing orientation, E and G grow large as the infill density increases. In terms of infill pattern, because the triangle is a more robust fill than lines, I expect the triangle pattern to have a comparatively higher E and G at all printing orientations. Our results (Figure 4.5) confirm this prediction: E and G are higher for all printing orientations with triangle infills *vs.* lines. Finally, for the printing orientation tests, it is well known that FDM printers create 3D models with anisotropic properties—models are stronger in the X and Y direction compared to the Z direction (*i.e.*, a tensile load orthogonal to the FDM layers is the weakest). As expected, Figure 4.5 shows that as the printing orientation shifts from 0° (rod is printed vertically) to 90° (rod is printed horizontally), the tensile strength increases. In terms of the printing orientation’s effect on E and G , we found that 45° results

in minimum values for both Figure 4.5.

B.3 Mechanical Experiment 2 Setup

I followed a similar procedure to Experiment 1 but without extensometers: springs were placed into the load frame grips and stretched at 30mm/minute. To fit the springs into the grips, we added two flat grip plates to the ends of our spring models (Figure 4.3b). As a spring is stretched, it begins to elastically deform—a state which is reversible. This continues until an elastic limit is reached—the *yield point*—a threshold where the spring is permanently deformed or can even break apart. As before, the load frame software recorded the applied load and the grip displacement at 10Hz. In addition, the software automatically marked the yield point. From this data, for each spring, we can calculate k using *Hooke's Law* and derive G from Eq. 4.1.

B.4 Mechanical Experiment 3 Setup

In this experiment, to twist the 3D-printed spring and measure torque, we added a base plate and a socket to the spring, which was connected to a *NEMA 23* stepper motor and a *FUTEK Model TFF400* torque sensor (max 1130 N-mm)—see Figure 4.3c. We incremented the stepper motor angle by 2.8125° at 1Hz and recorded the torque sensor at 10Hz. Tests concluded when either the spring buckled due to overtwisting or slipping occurred. For analysis, we used readings from the stepper motor, the torque sensor, and a protractor to measure rotation angles (Figure 4.3c).