Karlsruhe Institute of Technology

Carnegie Mellon University

Automated Design of Tendon-Driven Soft Foam Hands using Markov-Chain-Monte-Carlo Optimization Methods

Master's Thesis
of

Dominik Bauer

KIT Department of Mechanical Engineering
Institute for Automation and Applied Informatics (IAI)

KIT Department of Informatics
Institute for Anthropomatics and Robotics (IAR)
High Perfomance Humanoid Technologies Lab (H^2T)

CMU School of Computer Science
The Robotics Institute (RI)

Referees: Prof. Nancy Pollard
Prof. Tamim Asfour
Prof. Ralf Mikut

Advisors: Julia Starke

Duration: June 1st, 2018 – October 1st, 2018

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Sunnyvale, October 1st, 2018

Dominik Bauer

Abstract

Multi-fingered soft robot hands promise safe and robust grasping even in unstructured environments. This has fueled the interest in soft robotics and led to the development of a large variety of new designs. However, materials that can undergo large elastic deformations require the rethinking of conventional hand morphologies, actuation and control strategies. Therefore, this thesis presents an approach towards a more automated design process, specifically dedicated to the class of soft foam hands. To leverage the full potential of such tendon-driven robots, an end-to-end process that supports the creation of hands that achieve specific grasps and manipulations is proposed. Further contributions include the creation of hand morphologies with varying rest poses and the development of a Monte-Carlo-Markov-Chain optimization approach for finding efficient tendon-routings and contraction levels. It is demonstrated that this approach is able to scale to the high-dimensional design domain of soft foam hands and can even produce routings that outperform designs created by humans. The results achieved in this work promise that automated design methods can indeed improve the development of purposeful soft hands in the future.

Kurzzusammenfassung

Roboterhände aus weichen und nachgiebigen Materialien sind vielversprechend hinsichtlich eines sicheren und robusten Greifverhaltens, auch in komplexen ungeordneten Umgebungen. Dieser Sachverhalt hat in den letzten Jahren das allgemeine Interesse am Gebiet der Soft Robotik vergrößert und zu einer Vielzahl von neuentwickelten Designs geführt. Mit der Verwendung von Materialien, die extrem große elastische Deformationen durchführen können, gehen aber auch neue Herausauforderungen bezüglich der verwendeten Handmorphologie, Aktuierung und Regelung einher.

Der in dieser Masterarbeit vorgestellte Ansatz zielt auf einen automatisierten Designprozess von nachgiebigen Roboterhänden, speziell von weichen, schaumstoffgefütterten, sehnengetriebenen Händen ab. Um den sehr komplexen Parameterraum dieser Hände zu untersuchen und deren volles Potential auszuschöpfen, wird im folgenden ein vollumfänglicher Designprozess vorgeschlagen, welcher die Konstruktion von Händen ermöglicht, die spezifische Griffarten und Manipulationssequenzen durchführen können. Dieser Ansatz umfasst die Erstellung von neuen Handmorphologien mit unterschiedlichen Initialposen und der Entwicklung einer Markov-Chain-Monte-Carlo Optimierung, die Lösungen für effiziente Sehnenverläufe zur Aktuierung der Hände findet. Es wird gezeigt, dass diese Vorhergehensweise gut auf die hohe Dimensionalität des Problems skaliert und in der Folge sogar Lösungen gefunden werden, die von Menschen konstruierte Hände hinsichtlich Präzision bei der Griffausführung übertreffen. Die präsentierten Ergebnisse sind vielversprechend und zeigen, dass vollautomatisierte Konstruktionsprozesse in Zukunft zu einem gezielteren Entwicklungprozess von nachgiebigen Roboterhänden beitragen können.

Acknowledgments

First of all, I would like to thank Nancy Pollard for providing valuable academic guidance and for sharing her incredible experience and immense knowledge. Her encouragement helped me to reach another academic level. Furthermore, I would like to express my gratitude to Tamim Asfour. He always had an open door and was very supportive of my research interests. He gave me the opportunity to research in this exciting field and by that enabled me to achieve more than I could have imagined. In addition, I want to thank Ralf Mikut for sharing his expertise and for providing guidance throughout this thesis. He did not hesitate to agree on being my advisor and always had an open mindset for collaboration with others which was essential for this thesis. Furthermore, I would like to thank Julia Starke for reading all of this work and giving valuable comments and suggestions.

I am also grateful for the valuable advice and the insights received from Stelian Coros and James Bern. Also, special thanks to Jonathan King, Yuzuko Nakamura and Kai-Hung Chang for the great time I had working with them at CMU and for their help on bringing soft foam hands to life. Additionally, I want to thank Marcel Kost for his advice and for sharing his expert knowledge on compiler and hardware architecture.

Of course, I want to thank Cornelia Schlagenhauf, for countless fruitful discussions, for her infinite support in- and outside the lab and for never getting tired of me.

Last but not least, I want to thank my family and friends for always being supportive and for encouraging me to pursue my goals and dreams.

The research towards this thesis was conducted at the Robotics Institute at Carnegie Mellon University as part of the Continuous Learning in International Collaborative Studies (CLICS) program. I want to thank all responsible persons that dedicate their time to organizing such an important program. I hope future students will be given the same opportunity to improve professionally and personally like I did.

Contents

1. Introduction

Aging populations, demands for higher production flexibility and an increasing world population have led to a growing interest in intelligent autonomous systems that can support humans in their daily life at work and at home. The list of applications for such systems is long and ranges from human assistive systems for elderly care, to supporting workers with handling heavy objects in manufacturing and maintenance. Other applications include medical tasks such as surgery, search and rescue tasks or harvesting crops. In industrial applications, robotic systems have long been successfully applied for tasks such as loading of machines or spot welding. Reasons for this are that such tasks are highly repeatable and are usually executed within separate workspaces for humans and machines to avoid injuries. In contrast, future robotic systems will need to operate safely, robustly and adaptively also in complex unstructured environments to be applied more widespread. However, safe interaction with humans is difficult for traditional robots as they consist of hard and rigid links and joints.

An emerging class of robots that circumvent this problem are referred to as *soft robots* which are being made from intrinsically soft materials. Apart from being safe, the softness and compliance achieved by these robots can be exploited to reduce the complexity of interactions with the environment. For example most methods for in-hand manipulation rely on exact models of the hand and the manipulated objects. To manipulate unknown objects as well, robotic hands need to reactively adapt to the object. In the case of soft robots this adaptation is achieved by the material itself without the need of complex low-level control. Such an exploitation of compliance can be observed in many biological organisms and is therefore a promising characteristic especially with respect to solving dexterous grasping and manipulation. However the use of soft compliant materials introduces a number of intricate problems that need to be addressed in order to create robots that achieve their full potential. The key challenge hereby lies in developing controllable robots within the vast design domain that besides a large selection of materials poses virtually no restrictions on the robot's shape, actuation and control. To complicate this even further, the design of each sub-domain itself (material, shape, actuation, control) is predicated upon existing sub-solutions with all design parameters being highly interdependent. In addition to the lack of geometric constraints traditional computer-aided design (CAD) methods are generally unsuited to manually explore unconventional shapes and deformations. Even for experts, this largely inhibits the exploration of the design space and thus the development of purposeful soft robots that can achieve a desired behavior.

This necessitates the development of automated design algorithms that can fully explore and support the creation of soft robots. For the specific class of tendon-driven soft foam robotic hands, this poses the challenge of creating functional goals, defining a foam morphology and rest shape, and finding a suitable tendon placement for actuation.

In an ideal case this would lead to a design process in which a user simply specifies a functional task for which an optimal design would be determined computationally. However, even optimization of one sub-domain, e.g. actuation is computationally very expensive. Therefore current computational resources are not able to optimize the complete and unconstrained design domain of soft robots. Instead optimization algorithms commonly support the design of either one or two sub-domains under the constraint of the remaining domains which are specified by human input. The goal of this thesis to provide an automated design process for soft robots, specifically for soft foam hands. This novel class of tendon-actuated soft robots consists of a foam core and is introduced in detail in Chapter 3. Overall, this thesis presents a complete approach for automating soft foam robot hands and is organized as follows:

Chapter 1 gives an introduction to the problem, presents related work in soft robotics focusing on design and optimization and motivates the purpose of this work. Chapter 2 provides a review of fundamentals relevant for grasping and manipulation and presents different optimization techniques available. Chapter 3 describes the manual design and fabrication process of soft foam hands. The main contribu-

tions of this thesis are presented in Chapter 4 which details the entire optimization approach from task to design. A verification of the algorithm and experiments and results are discussed in Chapter 5. Chapter 6 summarizes the results and concludes this thesis with an outlook on future work.

1.1. Related Works

Over the years, the field of soft robotics has gained a lot of attention. Their soft structure makes them inherently safe and their compliant nature promises flexible motions with high adaptability. A variety of soft robots can be found in literature that rely on different materials, actuation and modeling strategies. This chapter introduces available concepts and focuses on how optimization of soft robots has been applied in related works.

1.1.1. Soft Robots

As they are inherently safe, soft robotic devices are predestined for physical interaction with humans. A variety of materials has been investigated and used in medical applications for rehabilitation [96], wearables [92, 95] or exoskeletons [80], toys [7, 6] but also for gripping and manipulation of objects.

The intended task and environment influence the choice of materials. Oftentimes the selection is based on the desired level of elastic deformation and stiffness. In terms of tensile strength (Young's modulus), materials used in soft robots are comparable to soft biological materials such as muscles, skin or cartilage [103]. Apart from using silicone elastomers, rubber or alginate to imitate the properties of biological materials, significant effort has been put into creating flexible sensors [55, 2] and artificial skin [91, 68].

Fluidic elastomer actuators (FEA) comprise the most commonly used actuation mechanism in soft robotics. A large body of research is dedicated to creating pneumatic FEAs [81, 129, 40] or artificial muscles [125, 88] for a variety of applications. Besides fluidic actuation, elastomers also come to use as active components in the form of electroactive polymers [93, 111].

Soft Robotic Manipulators

This section specifically reviews materials and actuation mechanism used in soft robotic grippers and manipulators. Many of these mechanisms can and have also been successfully used for locomotion, but are not included in this section. Shintake et al. [114] distinguish between mechanisms used for gripping by:

Actuation: A widely used mechanism for grasping that among others includes tendon-driven actuation, fluidic elastomer actuators and electroactive polymers. As this comprises the most relevant class of soft robotic manipulators for this thesis, more details are given below.

Controlled Stiffness: This mechanism is based on controlling the stiffness of grippers between a "soft" and a "stiff" configuration. First, while still soft, the gripper approaches the object and envelopes it. Then, its structure is stiffened resulting in a tight grip. Many variations of controlled stiffness grippers ranging from low-melting point alloys and shape memory materials to granular jamming [8] exist. A representative example of the granular jamming effect can be found in coffee grounds, which when loosely filled into a rubber bladder are free to move and can be put into any shape. However, evacuating air from the bladder will result in compression of the granular material forming a solid object that is preserving its shape.

Controlled Adhesion: These soft grippers take advantage of adhesion, which is the interface attraction of two surfaces. They can hold objects by generating large shear forces on the surface. Commonly known mechanisms for controlled adhesion either rely on dry adhesion [21], suction cups [1] or electroadhesion [113].

[1] *Festo Co.Ltd. Octopus Gripper*

A number of fluidic elastomer actuators have successfully been used in anthropomorphic hands and grippers. Deimel and Brock [22] developed a hand from reinforced silicone that is pneumatically actuated and capable of grasping. Their more recently developed hand [23] is even capable of dexterous grasping which they evaluate using the GRASP taxonomy from Feix et al. [31]. Marchese et al. [75] and Katzschmann et al. [60] present a soft planar manipulator that is able to grasp objects under uncertainty in terms of position and shape of the grasped object. By reducing the amount of gas needed for inflation Mosadegh et al. [81] developed a soft robot that is able to achieve high rates of actuation. Examples of further pneumatically actuated grippers are octopus-inspired and rely on tentacles that are able to hold a flower Martinez et al. [76].

Most tendon-driven soft robots available combine rigid links with elastics hinges that replace the mechanical springs found in traditional tendon-driven robots[34, 87, 73]. While this leads to a simpler system they cannot be considered truly soft due to the presence rigid materials.

Thus, more related to the work in this thesis are systems in which any rigid structure has been replaced with soft materials. Mutlu et al. [83] developed a soft monolithic finger that can be fabricated using additive manufacturing. Another tendon-driven approach inspired by octopus are the soft tentacle arms developed by Calisti et al. [11] which later served as basis for their soft-eight-arm *OCTOPUS* robot that is capable of grasping and locomotion under-water [17]. Some more unconventional materials have been used by Jeong and Lee [53] who developed an origami robot made from paper. Most closely related to the research on foam hands is the work by Bern et al. [7] who developed tendon-driven plush robots using textiles.

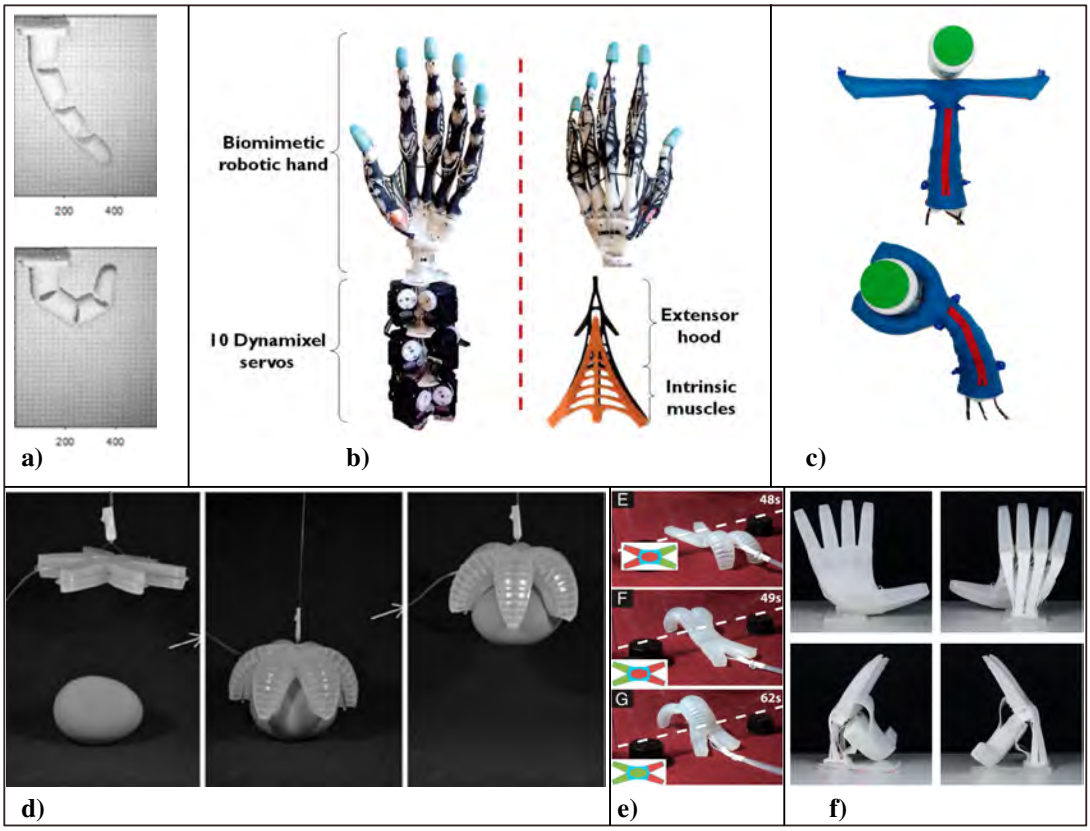


Figure 1.1.: Soft hands. Tendon-driven: a) 3D printed finger [83], b) biomimetic anthropomorphic hand [128], c) plush 2D gripper [7].
Fluidic actuated: d) starfish gripper [50], e) quadrupedal robot [112], f) RBO 2 Hand [23]. From [108].

The design of soft robotic grippers and manipulators is subject to a trade-off between properties such as cost, weight and dexterity. The large variety of technologies and materials found in literature attests to this difficulty of designing soft hands that excel in all features. Currently available systems provide solutions to specific problems focusing only to improve properties at the cost of others. With respect to building dexterous hands Xu and Todorov [128] built an biomimetic anthropomorphic hand that closely mimics the mechanics of a human hand. While this design is highly dexterous and can reproduce a large number of grasps from the Cutkosky taxonomy its complex design makes it inaccessible for non-experts. This is also the case with other highly dexterous designs [23]. In terms of weight, tendon-actuated elastomer grippers have the disadvantage of having a relatively high density making them heavy and dull. This problem is often circumvented by using FEAs which are lightweight, low-cost and can even be dexterous. However, a major drawback of FEAs, which is easily overlooked is the required compressor which takes up space, weight, and produces heat and vibrations. Additionally, leaking or damaged tubes and airchambers can easily result in a failure of the system. Apart from their lack of dexterity, this is also critical with grippers that rely on granular jamming.

Modeling and Control of Soft Robots

The benefits that arise from their continuously deformable nature come at the cost of soft robots being incredible hard to model and control. Solving kinematics is already a complex and difficult problem. Their ability to accomplish motions such as buckling, contraction, extension or bending, results in soft robots having virtually infinite degrees of freedom. Additionally, non-linear material effects such as compliance and hysteresis account for the non-trivial nature of this problem [36]. The wide range of design and actuation techniques found in literature further complicates this circumstance as it is impossible to distill a universal model for all robots. Previous works have particularly studied the problem of inverse kinematics (IK) [101, 120, 54]. Existing control approaches can be classified into three main categories: model-based controllers, model-free controllers and a combination of both.

Model-based controllers rely on the establishment of a kinematic model from which the actuation can be directly inferred for the desired configuration. Saunders et al. [106] model caterpillar-like soft robots as a series of extensible linkages. For tentacle-shaped soft robots Marchese et al. [75], Marchese and Rus [74] and Chen et al. [15] use piecewise constant-curvature models to model the robot. For soft robots with arbitrary shapes work by Duriez [25] presents a real-time solution using a finite element method (FEM).

Model-free approaches offer a wide variety of data driven techniques to control soft robots. Neural networks have successfully been used learn inverse kinematics on a cable-driven soft tentacle manipulator with 2 degrees of freedom [38]. Rolf et al. have proposed an exploration algorithm for creating task space samples for IK learning [101].

1.1.2. Design and Optimization of Morphology, Materials and Control

This section highlights related works which successfully utilized optimization techniques to improve the performance of robots with a special focus on soft materials and designs.

Although optimization has already been applied to improve traditional robot designs, the unconstrained, interdependent design space of soft robots raises additional questions about morphology, materials and controlling motions that can best be answered by the application of automated design techniques.

Deimel et al. [24] propose a co-design method that simultaneously optimizes morphology and control of a pneumatically actuated soft hand. In order to find morphologies that complement the control parameters and vice versa they use a particle filter optimization method which searches both domains for a set of optimized hand and control parameters. Sampled particle solutions are evaluated in a simulator by rolling out a grasp sequence and judging whether it was successful or not. They specifically show that their optimization approach scales well to high-dimensional parameter spaces, but limit the domain to a relatively simple grasping scenario.

In order to evolve soft robots Rieffel et al. [99] explore three different strategies of finding interdependent solutions in terms of morphology, material and control parameters. Their strategies follow the

general approach of holding one property constant while co-evolving the other two. The first strategy is considered with placing actuators for locomotion on a caterpillar robot with ten arbitrarily placed tendons while optimizing for the respective control patterns. They encode each solution as a genom consisting of the attachment points of each tendon and the respective activation steps. A genetic algorithm then progressively optimizes the gait for a number of generations before fixing the best found activation pattern and evolving the attachment points of the tendons. Their second strategy investigates the interplay of control and material properties for a fixed morphology using a similar encoding. In their last approach Rieffel et. al. use a generative encoding [45] to explicitly generate body shapes while implicitly determining material properties as well as muscle placement [99].

Other research concerned with optimizing material parameters has been introduced by Hiller and Lipson [44], who obtain locomotion of their designs by finding heterogeneous material distributions with the help of evolutionary algorithms.

Finding inspiration in animals for designing soft robots is an approach that does not rely on complex algorithms. Many bioinspired worm or caterpillar-like robots can be found in [62].

Inouye and Valero-Cuevas [51] optimize anthropomorphic tendon-driven robotic hands focusing on improving the grasp quality of the designs and show that robotic hands can even exceed human grasping. They evaluate the performance of a design in terms of the achieved grasp quality for a precision grasp. In order to optimize tendon-routings they use an MCMC approach that accepts new tendon layouts based on whether a grasp can be maintained against an external force or not. Their work specifically emphasizes how optimization of robotic hands produces designs that significantly outperform non-optimized designs.

1.1.3. Challenges and Contributions

Many interesting studies explore new materials, geometries, fabrication techniques and actuation mechanisms exist. However a safe, widely-accessible soft hand that is capable of dexterous manipulation is not available. This emphasizes the demand for low-cost and customizable hands as opposed to more general-purpose designs already available. Soft foam hands, explained in detail in Chapter 3, are a new class of soft robots that significantly differ from previous systems and offers an unbounded multi-fold design space:

- Tendons can be placed anywhere on the foam to achieve task specific actuation.
- There are no restrictions in terms of possible foam shapes and morphologies. Grippers, anthropomorphic hands and even multi-fingered non-anthropomorphic hands can be designed and fabricated.
- All materials used are readily available and accessible for non-experts.

The nature of this design space raises several questions that need to be addressed in order to provide potential users with an end-to-end pipeline that describes design, fabrication and control.

A new methodology for fabricating foam robots has been developed and described in [108]. In terms of modeling and control, FEM simulations and data-driven model-free approaches have previously been applied to model soft robots. However not just one soft foam robot with a predefined morphology needs to be controlled, but potentially infinite amounts of possible tendon routings. This requires a mapping from pose to actuation with emphasis on data efficiency, models that scale well to new geometries or tendon routings and an intuitive direct control. Possible strategies have been investigated and compared in [109].

With respect to the design of foam robots, questions about shape, actuation and controls need to be answered. Related works have highlighted the effectiveness of optimization for this purpose. Nevertheless, existing approaches in literature either limit the design domain to 2 dimensions [51] and simple grasping scenarios [24] or provide optimization for rather simple caterpillar-like robots [99] that do not compare to the complexity of hands and grasping. This suggests that a general optimization approach that captures the true complexity of grasping and manipulation for any object shape and property remains largely out of reach.

To address this, two concurrent approaches are being followed. Ideally, designing foam hands should be an fully automated process. However, in traditional design processes users need to explicitly design hand shapes and routings based on intuition in order to create hands that can achieve specific grasps. Unfortunately, even with todays computational power it is incredibly difficult to create algorithms that can optimize all properties at once for such largely unconstrained domains. This constitutes the lack of automated design tools available and motivates the first approach that is pursued. Instead of automating the process the goal is to provide users with tools to support their intuition, specifically enabling them to quickly explore different effects of shape and actuation in simulation. This approach is described in detail in [108]. The long term strategy presented in this thesis envisions a more functional approach, where users merely specify goal grasps or motions and an optimal design in terms of morphology and control is automatically generated. This works makes the following contributions towards this goal:

- Provide a universal shape-independent methodology to specify goal grasps and motions
- Establish an approach that generates foam hand morphologies with different rest poses that are within kinematic constraints of the problem
- Develop an algorithm that optimizes tendon routings and activations for a fixed morphology and rest pose
- Provide a proof of concept that dexterous grasping and manipulation can be optimized for
- Demonstrate that optimized tendon-routings can achieve precise posing that even outperforms human designed routings

2. Fundamentals

2.1. Grasping

Despite decades of research dedicated to improving performance and dexterity, robotic hands still fall short of the human hand. Especially in unstructured and cluttered environments robots are not able to sufficiently interact with objects, tools and humans due to their inability to grasp and manipulate. Reasons for this are manifold and include the lack of robust hardware, sensors and a limited understanding of how to control hands in the wild. In order to achieve similar levels of dexterity and flexibility the human hand has been subject to extensive studies and often serves as inspiration for robotic hands. This section summarizes the most relevant fundamentals on human and robotic hands.

2.1.1. Human Hand Anatomy

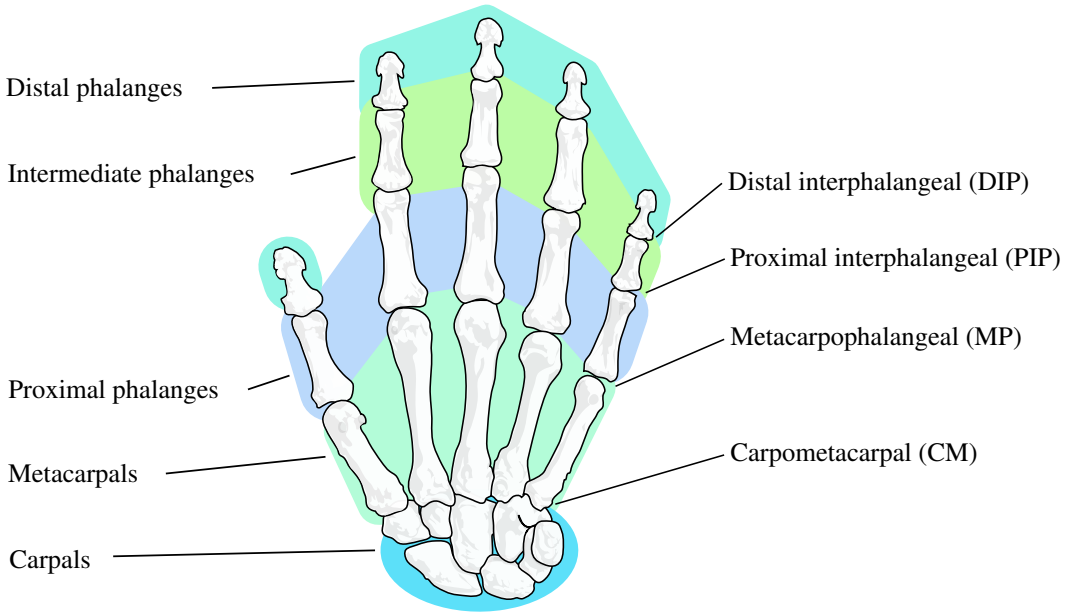


Figure 2.1.: Bones (left) and joints (right) of the human hand.

The Human hands consists of a wrist, a palm and fingers. It has 27 degrees of freedom (DOF) and 27 bones which can be organized into five different groups shown in Figure 2.1 on the left. The wrist contains eight carpal bones which connect the hand to the lower part of the forearm. The five metacarpal bones that extend from the wrist to the base of the fingers form the palm of the hand. The remaining 14 bones are called phalanges and form the four fingers and the thumb. Each finger consists of a proximal phalanx, intermediate phalanx and a distal phalanx. The thumb only consists of a proximal and intermediate phalanx. In order to describe movements and motions this work uses the anatomical terms flexion,

extension, abduction and adduction which are defined as follows:

- **Flexion:** a bending movement around a joint that decreases the angle between the links at the joint
- **Extension:** an unbending movement around a joint that increases the angle between the links at the joint
- **Abduction:** a motion that moves fingers away from the hand's mid-line
- **Adduction:** a motion that moves fingers toward the hand's mid-line

Depending on the application there is a large variety of human hand models available that use different kinematic models or have more or less degrees of freedom [41, 94, 19].

2.1.2. Grasp Taxonomies

In order to understand grasping many attempts have been made to classify different types of grasps and incorporate them into a single distinct taxonomy. From a neuroscience perspective grasp taxonomies are helpful for understanding human grasping, in terms of robotics they can serve as inspiration for robot hand design, benchmarking designs and grasp recognition [28]. One of the first attempts to characterize human grasping was done by Napier [85] who introduced the distinction between *power grasps* and *precision grasps* (see Figure 2.2). This classification can be found in many taxonomies and refers to the idea that the "nature of the intended activity finally influences the pattern of the grip" [85]. In the case of power grasps the relation between object and hand is rigid and the combined fingers form a clamp against the palm. Precision grasps are characterized by one or more finger tips being in contact with the object. This configuration allows for intrinsic movements of the object.

Figure 2.2.: Left: Power grasp of a ball. Right: Precision grasp of a small sphere. From [31].

Based on the broad classification done by Napier, various taxonomies that further subdivide grasps have been developed. One of the most cited taxonomies is that of Cutkosky [20] who observed machinists during their work and divided the grasps into a hierarchical tree of 16 types. This subdivision is based on the constellation of fingers that exert force on the object and on contact points with the object. As depicted in Figure 2.5 object size and power of grasp decreases from left to right. Kamakura et al. [57] classify static prehensile patterns in terms of purpose of grasp, hand shape and contact areas with the object.

Considering grasps from previous taxonomies and from different domains, Feix et al. [31] rearrange a wide set of grasps and organize them into 33 grasp types. They arrange them in their *GRASP* taxonomy using the following four criteria:

- **Opposition type**:
  1. **Pad opposition** as depicted in Figure 2.3 a), contact occurs between hand surfaces generally parallel to the palm along the x-axis in Figure 2.3 d).
  2. **Palm opposition** as in Figure 2.3 b) occurs between hand surfaces generally perpendicular to the palm along the z-axis in Figure 2.3 d).
  3. **Side opposition** as shown in Figure 2.3 c) occurs between hand surfaces generally transverse to the palm along the y-axis in Figure 2.3 d).

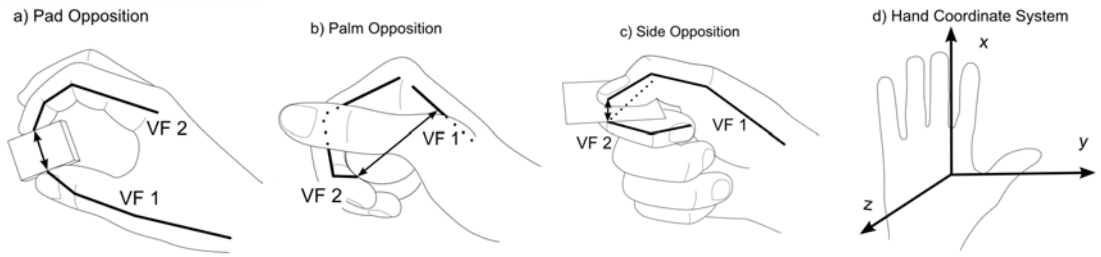


Figure 2.3.: Opposition types and virtual fingers (VF) for different grasps. Figures from [31]

- **Power, Intermediate, Precision Grasps**: The idea of power and precision was adopted from Napier [85] and Landsmeer [66]. For finer differentiation between power and precision, intermediate grasps were later added by Kamakura et al. [57] and Skerik et al. [117].
- **Virtual finger assignment**: The concept of virtual fingers refers to the fact that in many tasks several fingers form a functional unit and apply forces in similar directions. To simplify grasps one or more fingers and even parts of the hand can be assigned to a virtual finger. In Figure 2.3 b) the palm serves as virtual finger (VF) 1 while the opposing fingers are assigned to VF 2.[49, 31]
- **Position of the thumb**: A new feature introduced in their taxonomy. In this thesis the position of the thumb played an important role for designing new foam hands and especially for selecting suitable restposes. In the *GRASP* taxonomy it is distingushed between abducted and adducted thumb CMC Joint.

Nancy Pollard[1] regroups the grasps from the GRASP taxonomy into the following six categories:

- Power grasps using the palmar gutter
- Power grasps using other parts of the palm
- Power grasps with lateral stabilization
- Precision grasps with lateral stabilization
- Power grasps with pad opposition
- Precision grasps with pad opposition

This approach considers grasp type (power/precision) and which parts of the hand are used to hold the object. The six groups and the respective grasps are shown in Figure 2.4, which also includes the number each grasp is assigned to in the GRASP taxonomy by Feix et al. [31].

[1] `http://graphics.cs.cmu.edu/nsp/index.html`

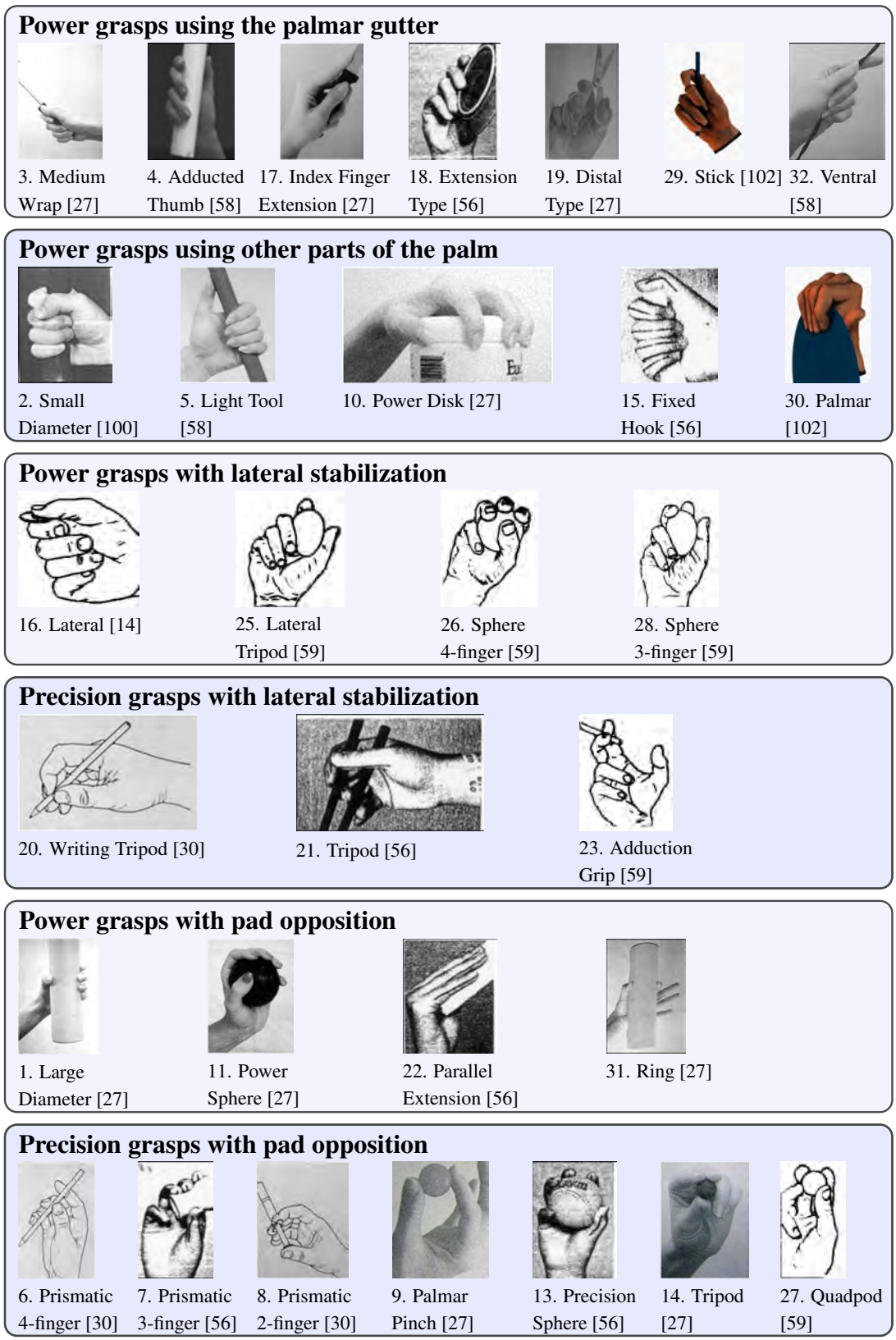


Figure 2.4.: Cumulative taxonomy. Grasps are divided into six groups considering power/precision type and used hand parts.

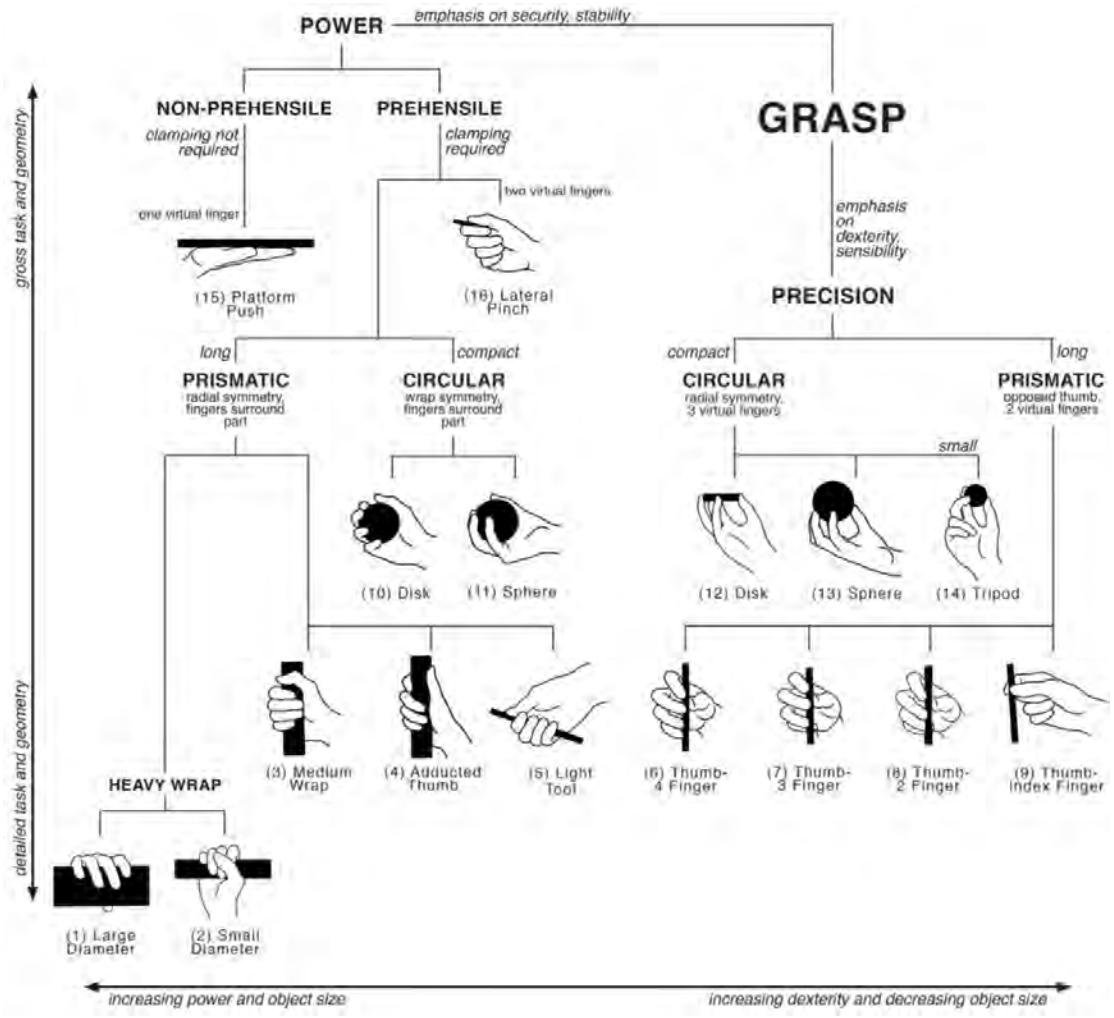


Figure 2.5.: Cutkosky taxonomy. Grasps are divided into a hierarchical tree, where from left to right, object size and power of grasp decreases. Retrieved from `https://hackaday.io/project/9890/logs`.

2.2. Manipulation

The ultimate goal for robotic hand research is to build dexterous humanoid hands that can successfully interact with cluttered and unstructured environments. One prerequisite for that is that robot hands can perform stable and reliable grasps. In real scenarios this seldom is enough. Humans and primates use different non-prehensile strategies like pushing, rotating, sliding or bending objects to achieve their goals. For robot hands this means that in addition to performing various types of grasps, they need to be able to manipulate objects in order to be successful. This unfortunately is a sophisticated task for robots, as apart from unknown object properties (geometry, mass, surface roughness, compliance, etc.) the robot also requires a continuous state estimation of the manipulated object. In addition to the robot's own state estimation this makes manipulation a very hard problem. Therefore, current robot interaction with objects is mostly limited to locating an object, grasping it and placing it somewhere else (*pick-and-place*). In order to capture and identify possible manipulation actions there have been many attempts in the literature to define a taxonomy. Focusing on pre-grasp interactions with objects, Chang and Pollard

[12] classify object adjustments into rigid transformations and non-rigid reconfigurations. They define rigid transformations for example as rotating a cup by its handle before grasping it. An example for non-rigid object reconfiguration is curling a piece of paper to achieve a pinch grasp. Elliott and Connolly [29] group manipulations into three classes of intrinsic movements, distinguishing between simple synergies such as squeezing, reciprocal synergies such as rolling and sequenced patterns of movements which are used to change contact points of fingers on the object. In order to define a taxonomy that applies to any hand and focuses on what the hand is doing during execution of manipulation tasks, Bullock et al. [9] choose a hand-centric view on the problem. Their proposed taxonomy consists of small sets of yes/no criteria which define for example whether the hand is in contact, the task is prehensile or not, or if there are relative motions between hand and object at contact. Recent work by Nakamura et al. [84] analyzes several high-speed captures of motion sequenzes and uses different manipulation taxonomies to classify their observations. The motions were classified using the static grasp pose taxonomy by Feix et al. [31], the intrinsic manipulations by Elliott and Connolly [29] and the more high-level taxonomy by Bullock et al. [9]. Additionally they also focused on errors, recoveries and on the importance of contacts. Even with their very broad approach of classifying manipulation sequences, Nakamura and her colleagues [84] find "that the process of grasping is complex [...] particularly in situations with clutter and environmental constraints." This suggests that additional research is required to fully understand human grasping strategies in the wild. More general taxonomies that capture the full array of human manipulation are much needed to help and guide implementation of new control strategies for robots.

2.3. Kinematics

Kinematics studies the motion of bodies without considering masses or forces that act on it. Robot links can be modeled as rigid bodies that are connected by joints and form a kinematic chain, which constrains the possible degrees of freedom. In robotics, kinematic models are generally used to describe the relation between joint angle configuration and position of the robot end-effector (also known as tool center point - TCP). This yields two problem formulations depending on whether the goal is to determine the end-effector position given the configuration of the joints (*forward kinematics*) or whether the goal is to find a configuration of joint angles for a given TCP pose (*inverse kinematics*). For rigid-body robots both problems are well studied and thus will not be further explained in this thesis. Detailed fundamentals on kinematics can be found in [82]. Looking at the typical grasp synthesis with known object geometry and location it is obvious that from a kinematics perspective grasping an object is a matter of finding the correct joint angles that lead to the desired TCP position. As discussed above, for rigid-bodies this problem can be easily solved. For soft robots however, this problem becomes significantly harder. This is due to their compliant structure which cannot be modeled as a linked chain of rigid bodies or because they do not even have joints which is the case for soft foam robots used in this thesis. Section 1.1.1 gives a more detailed look on how soft robotic hands are designed and modeled and discusses different approaches found in literature to solve the IK problem.

2.4. Finite Element Simulation

In general, the finite element method can be used to approximate desired quantities of continuous structures by discretization of the structure into basic geometries. Depending on the application FEM can be used to model a large variety of physical properties, e.g. the displacement of mechanical structures, the velocity of flows, the temperature in heat conduction problems or the electric potential in electrostatics. In this thesis FEM is used to model the continuously deformable nature of soft robots which is a a popular approach [26, 67]. The general procedure for applying FEM is the same independent of the modeled properties. This section focuses on approximating the displacement of mechanical structures as this forms the theoretical basis on which the simulation framework for soft foam robots is based on. The basic idea of FEM is built around minimizing an energy functional. This energy functional E combines all energies that can be associated with the finite element system. In the case of foam robots the total

energy consists of the deformation energy in the foam and the strain energy stored in the tendons. In order to find the minimum of the energy functional the derivative of the functional with respect to the deformation needs to be set equal to zero : $\frac{\partial E}{\partial u} = 0$. Although each problem is different and requires the selection of appropriate material models and boundary conditions, the general process of applying a finite element analysis is the same. According to [98] it involves the following steps:

1. **Mesh generation/Discretization**: This step is characterized by approximating the continuous domain by discretizing it into a collection of predefined elements. There is always a tradeoff between finer meshes that yield better approximations and coarser meshes which are computationally faster.

2. **Derive element equations for all elements**: Obtain element equations in the form of $[\mathbf{K}_e]\ \vec{u}_e = \vec{F}_e$. and derive element interpolation functions.

3. **Assemble element equations to obtain overall system equation**: From the sum of all element equations, the overall system equation can be derived as $[\mathbf{K}]\vec{x} = \vec{F}$ with $[\mathbf{K}] = \sum_1^n [\mathbf{K}_e]$.

4. **Identify boundary conditions of the problem**: The global primary and secondary degrees of freedom need to be identified.

5. **Solve the assembled problem**

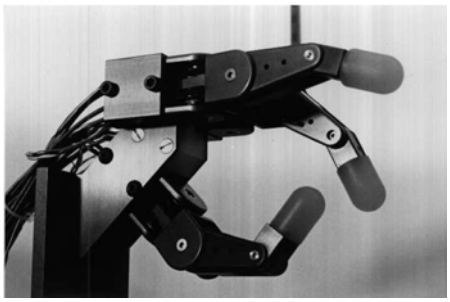
6. **Postprocess the results**

The fundamentals of finite element simulation will not be discussed further in this work as this is a well-studied problem, a detailed introduction can be found in [98]. In the special case of modeling soft foam robots with FEM, large deformations and non-linear material behavior have to be considered. The exact approach of how soft foam robots are modeled in this thesis builds on work from Bern et al. [6] and is explained in detail in Section 3.1.2.

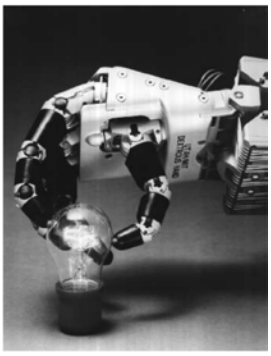
2.5. Robotic Hands

Ever since the word *robot* has first been introduced in 1920 by Czech playbook writer Karel Capek there existed a great inspiration of building machines that can execute tasks in a human-like fashion. Aside from the development of rather functional grippers that found application in industrial manufacturing a large number of multi-fingered hands has been developed in the last 45 years. Such hands have especially been aiming at achieving similar human levels of dexterity and manipulation capabilities. One of the early tendon-driven designs featuring 3 fingers, able to perform tasks such as attaching a nut to bolt was developed by Okada [89]. Two major multi-fingered hands that have been developed in the 1980s are the Salisbury Hand [105] and the Utah/MIT hand [52]. Moving away from tendon-driven concepts a number of hands have been developed around the year 2000 that use mechanical linkages instead of cables for actuation (DLR Hand [10], GIFU Hand [61], Robonaut Hand [72]). More recent robotic hand developments such as the Universal-Hand [47], the Shadow Dexterous Hand[2] or the DLR/HIT [71] are lightweight five-finger designs of human hand size.

[2] https://www.shadowrobot.com/products/dexterous-hand/



a) Salisbury hand. From [82]

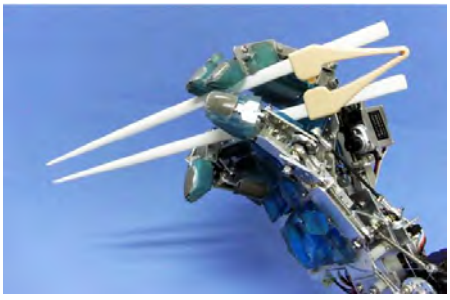


b) Utah/MIT hand. From [82]

c) DLR hand [3]

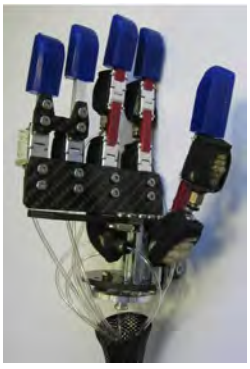
Figure 2.6.: Early designs of multi-fingered robotic hands.

Apart from tendon-driven and direct actuation mechanisms, hands such as the TUAT/Karlsruhe Humanoid Hand [32, 33] utilize smart under-actuation mechanisms that allow the hand to automatically adapt to different grasp shapes and forces without having to rely on sensory feedback. In terms of prosthetic hands Schulz et al. [110] developed a hand that closely approximates the grasping abilities of the human hand. Their five-finger hand has 15 degrees of freedom and is driven by flexible fluidic actuators which enables reliable grasping of a large variety of objects. Another example of a hand actuated by flexible fluidic actuators is the FRH-4 hand [35] developed for the humanoid robot ARMAR-III by Asfour et al. [4].



a)TUAT/Karlsruhe humanoid hand. From [33]

b) Karlsruhe prosthetic hand. From [110]



c) FRH-4 hand. From [35]

Figure 2.7.: Different multi-fingered robotic hand designs developed at Karlsruhe Institute of Technology

2.6. Optimization

Optimization can be described as the process of finding maxima or minima of a given function. Since many problems in engineering, computer science and even economics can be formulated as optimization problems, a large area of research across all domains is dedicated to developing new methods and algorithms. This section first gives a broad overview of the field of optimization, introduces the most

[1] https://www.dlr.de/rm/en/Portaldata/52/Resources/Roboter_und_Systeme/Hand/Hand_I/hand_I.jpg

important fundamentals and then focuses on a more detailed explanation of methods that are applied in this thesis to optimize soft foam robots.

2.6.1. Local Optimization

One of the classical optimization techniques dates back to Lagrange [64], who developed the method of Lagrange multipliers which can be used to find local maxima or minima of a function $f(x,y,...)$ subject to the constraint of $g(x,y,...) = c$. In contrast to Lagrange's calculus-based approach, Newton for example developed an iterative method know today as *Newton's Method*. Most traditional engineering problems are concerned with solving a linear equation system of the form $Ax = b$. The classical solution method for such problems is commonly known as Gaussian Elimination and can be implemented as $LU - Factorization$ (LU referring to the decomposition of A into a *lower* and *upper* triangular matrix). Another popular method to decompose matrix A is known as the Cholesky-Factorization which is a computationally more efficient factorization but has the limitation that it only works for positive-definite matrices A. Forward substitution and back substitution of the decomposed system of linear equations leads to the direct solution of the problem. Therefore solvers using such techniques are commonly known as direct solvers.

Today, due to the advances in computational resources and their efficiency in terms of memory, iterative methods are especially popular for solving large linear equation system. As for direct solvers, the performance and choice of method hereby largely depends on the problem and resulting properties of matrix A. Most standard discretizations of partial differential equations (PDE's) for example result in sparse matrices where the term sparse refers to the few non-zero elements present [104]. In case the non-zero elements follow a structure the advantages in terms of storage capacity can be significant because only nonzero elements need to be stored. This property is commonly exploited when solving the assembled problem equations of finite element systems. Most popular iterative methods are the conjugate gradient or the bi-conjugate gradient methods [118].

While for convex problems finding the minimum (or maximum) is simply a matter of finding the lowest possible function value, this is not true for non-convex problems. Unfortunately, modeling accurate real-world problems oftentimes requires the introduction of non-convex objectives or constraints. For objectives with many local minima, local search methods cannot guarantee the optimality of the current solution because they only test for optimality within a local region. This means that local search methods can terminate in different local minima depending on the initial starting point of their search. This is not helpful in most cases, where a deterministic behavior that yields the same solution independent of the initial starting point is desired. Methods that can satisfy such requirements are called global optimization methods and are introduced in the following sections.

2.6.2. Fundamentals of Optimization and Mathematical Programming

It is assumed that most of the terms and definitions described in this section are known and are only mentioned here for the sake of completeness. An exhaustive work on global optimization can be found in Horst and Pardalos [46]. In terms of mathematical programming Sauppe [107] provides a maintained glossary which contains all terms related to this field.

Since $max(f(x)) = -min(-f(x))$ the problem of finding a maximum can always be transferred into finding a minimum. Therefore, the terms maximum and minimum are used interchangeably throughout this thesis. What makes solving such optimization problems so hard is that oftentimes there is no clear representation of the objective function f. So, when it comes to solving a *difficult* optimization problem, the true optimal solution is rarely found within acceptable time or memory. A good indicator on how difficult a problem is to solve with a certain algorithm is the time and memory complexity. With an increasing number of input variables the computational complexity increases. In computer science this relation is expressed in the Big-O notation (also known as Bachmann-Landau-Notation [5, 65]) which provides a bound on the maximum growth rate of functions depending on their input size. This dependency can be used to divide problems into *complexity classes*. While a large amount of research is considered with *computational complexity* and an incredibly large number of classes exists [1], only the

two most important ones, P and NP problems are introduced here. P-Problems can be solved within polynomial time by a deterministic Turing machine. According to Cobham's thesis this means that P-problems are the only class of problems that can efficiently be solved computationally [63]. In contrast to that NP-Problems are problems that are solvable in polynomial time only by a non-deterministic Turing machine (NTM) [90]. In theory an NTM is a machine that can select subsequent states from a set of multiples for any given state. Furthermore, it is not possible to infer the next state of the machine based on the input and its current state, which gives it the name non-deterministic. The class of $NP-complete$ is a subset of NP and contains problems for which there has not been found an algorithm that is able to solve it in polynomial time on a deterministic machine. Whether an algorithm exists such that $P = NP$ is one of the big questions in computer science. A selection of $NP-complete$ problems can be found in Section 2.6.3. In order to approximate a quasi-optimal solution for such problems in an acceptable amount of time, stochastic optimization techniques (see Section 2.6.4) become essential.

The challenges with $NP-complete$ problems are manifold. Oftentimes the objective function of such problems is not continuous, not differentiable and is highly multimodal (multiple maxima or minima). In addition, many real-world problems are subject to multiple objectives. Frequently, these objectives are contradictory meaning that it is impossible to find a single optimal solution (set of parameters) that maximizes all objectives at once [124].

Instead, a solution candidate is considered to be *Pareto-optimal* if there exists no other feasible solution which would improve one parameter while not simultaneously causing a worsening of another parameter. Hence, for multi-objective problems it is desired to find the set of *Pareto-optimal* solutions also known as *Pareto-Front*. Unlike in Figure 2.8 which depcits a convex *Pareto-Front*, many of the difficult features found in single-objective functions, such as multi-modality or discontinuity are also present in multi-objective problems.

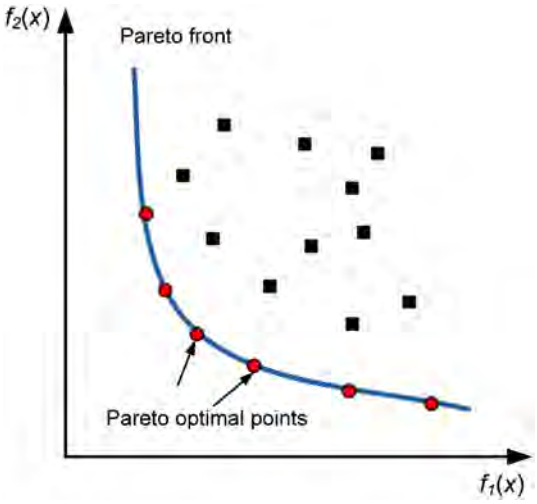


Figure 2.8.: Depiction of the Pareto Front for an optimization problem with two objective functions. In terms of robots $f_1(x)$ could hereby refer to the time required to execute a trajectory while $f_2(x)$ could stand for the required energy to execute the trajectory. From [48]

There are many other problems associated with applying optimization algorithms which have not been discussed. One problem that often occurs with highly multi-modal objective functions is premature convergence. This term refers to algorithms terminating early because they become stuck in local minima from which they cannot recover. Further problems arise with dynamically changing objective functions since the fitness of solutions depends on the timestep t. Solutions that are optimal in step t might not be optimal in $t+1$, which means that algorithms need to be adapted to be able to tackle such problems. As always when dealing with optimization algorithms that involve the evaluation of objective function values with the help of training data samples one can observe the phenomena of *overfitting* and *underfitting*.

To conclude, there are many difficulties associated with optimization and there is no algorithm that can find an optimum to all problems. Instead there are algorithms that perform reasonably on a number of problems or specialized algorithms that perform excellent on a single problem. This circumstance has been expressed by [127, 126] in what is commonly know as the *No-free-lunch-theorem*.

2.6.3. Problem Classes

Challenges that arise when solving *difficult* problems have been introduced above. This section introduces some typical $NP-complete$ problems that require the application of global optimization [86]:

Packing Problems

Packing problems involve packing a number of known k-dimensional objects into larger regions of the known k-dimensional space so that there is no overlap between objects and an additional cost critera (e.g. weight) is minimized. Many packing problems occur in real-life scenarios. A famous example is the *knapsack* problem where given a number of N items with weights and values, one has to place a maximum number of items into a container whilst maximizing the value.

Routing Problems

This class is concerned with finding an efficient path along a complex graph. Nodes represent locations and edges connecting the nodes have weights that represent the cost affiliated with traveling along the edge. There are two main types of routing problems depending on whether the goal is to visit a set of nodes or a set of edges. The most famous routing problem in computer science is the traveling salesman problem (TSP).

The name of this problem refers back to a time where salesmen had to travel from door-to-door to sell their goods. Given a set of locations (e.g. cities or addresses) the salesman's goal was to find the shortest route that visits all locations. For larger problem sizes this task is incredibly hard to solve as the number of permutations for n locations is $n!$. This means that for as little as 20 locations an exhaustive search of all possible paths is impossible. TSP problems can be modeled as graph with vertices $\{A,B,C,D,...\}$ representing the locations to visit and weighted edges containing the distance as depicted in Figure 2.9. In addition to the traditional TSP a variety of more general problems with asymmetric costs along edges exist. Solving TSPs is often used to benchmark the performance of new optimization algorithms. TSPs are an extensively studied class of problems offering various heuristics to approximate solutions. However, no solver can provide optimal solutions for all problem sizes and variations.

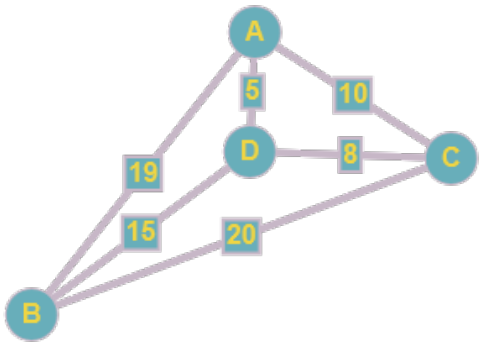


Figure 2.9.: Example of a graph consisting of nodes (A,B,C,D) and costs(e.g. distance) along edges connecting two nodes.

Scheduling Problems

The goal of scheduling problems is to match or assign a group of agents to a set of tasks so that the overall cost is minimized. Each worker is assigned with a fixed cost for the respective task. In the most basic case, called the linear assignment problem, the number of agents is equal to the tasks and the total cost of the assignment for all tasks is equal to the sum of the cost for each agent. To give a real-life example, the assignment problem is common in ride-sharing services. There are a number of drivers and a number of customers that need to be picked up and dropped-off somewhere. Since ride-sharing services want to minimize pickup times for customers and limit the time where cars are driving without a customer, they need to find the most efficient way to assign drivers to pickup customers. This can get increasingly complex considering that cars are differently sized and can transport 4 or more people and that maybe groups of people want to be dropped off or picked up along the route.

To provided some additional examples from the field of robotics [86]:

Trajectory Planning: Finding collision and obstacle-free trajectories along the robots workspace.

Control: PID-controllers are very popular in robotics. Although their structure is simple and they provide excellent closed-loop response characteristics it can be very challenging to tune PID param-

eters. Therefore various methods exist to tune these parameters using global optimization techniques [16, 13].

Design: Due to the amount of available parameters robot design is a complex problem. Topics that can be addressed span from defining appropriate link lengths for a specified workspace to assigning basic mechanical patterns like joint type and amount of joints.

2.6.4. Global Optimization

The problems presented in Section 2.6.3 are exemplary to a large number of other real-life problems for which it is interesting to find the global optimum. Due to the wide range of applications in engineering, science and economics a vast amount of methods exists today. In a broad sense the field of global optimization can be further divided into two subfields: deterministic optimization and stochastic optimization. While deterministic approaches such as *Branch and Bound* have been proven to be very effective in finding exact solutions, they are not able to scale effectively to large problems. Therefore deterministic approaches will not be further discussed here, more details can be found in [69]. This thesis will focus on the class of stochastic optimization, which does not guarantee optimal solutions but approximates the optimum in a reasonable amount of time. Especially popular with stochastic optimization techniques is the class of metaheuristics. Some key ideas and algorithms are introduced in the next section.

Stochastic Optimization and Meta-Heuristics

In general, randomness can be introduced into optimization in two different ways. Whenever measurements are involved, the input data of algorithms is subject to some level of random noise. This is especially relevant in real-time state estimation and control problems where noisy measurements are used to approximate the steady state of the system. Opposed to that randomness can also be injected into the search itself. Algorithms that rely on random sampling of data are known as Monte-Carlo Methods. Hartke [42] defines following aspects for global stochastic optimization methods in which they need to excel in order to be successful:

- Quickly find the local minimum for any initial starting point
- Escape the local minimum again
- Focus on the overall structure of the objective and do not get lost in irrelevant details
- Find a balance between exhaustively searching all minima and randomly exploring unknown regions (exploitation vs. exploration). Information collected in previous steps should be used to guide the search.

Stochastic optimization methods that are specifically designed to fulfill these requirements are called meta-heuristics. Today, a wide variety of meta-heuristics can be found in literature and new algorithms are still being proposed. There are many ways to classify meta-heuristics, e.g. single-solution vs. population-based, or nature-inspired vs. non nature-inspired approaches. A popular single-solution approach is *Simulated Annealing*, which improves and modifies only one single solution at a time. In contrast, population-based approaches such as evolutionary algorithms or particle swarm optimization use populations to guide the search. As suggested already by their names, many meta-heuristics are inspired by processes that can be observed in nature and try to imitate such processes to guide the search. While only the most relevant meta-heuristics for this thesis are introduced in the following an exhaustive database of currently existing meta-heuristics can be found in [39].

Markov-Chain-Monte-Carlo

As discussed in Section 2.6.3 typical objective functions of real-life problems oftentimes depend on a large number of parameters and can be discontinuous and unbounded. This makes calculating approximate gradients very expensive or even impossible. Whenever the set of feasible solutions in optimization is too large to exhaustively compare all solutions, Markov-Chain-Monte-Carlo (MCMC) approaches can be applied to sample candidate solutions efficiently [3]. The key ideas of MCMC methods are, as their name suggests, based on Markov chains and Monte Carlo methods. A Markov chain X can be formulated as a sequence of states X_n for which the probability of moving to the next state X_{n+1} only depends on the current state and not the previous states $X_0, X_1, \ldots, X_{n-1}$. Mathematically, this can be expressed as

$$P(X_{n+1} = x | X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n) = P(X_{n+1} = x | X_n = x_n)$$

where $P(.|.)$ denotes a conditional probability [37, 3]. The original idea behind Monte-Carlo approaches was to use randomly generated samples to compute integrals. Given an integral of the form

$$\int_a^b h(x)dx \tag{2.1}$$

$h(x)$ can be decomposed into the product of a function $f(x)$ and a probability density function $p(x)$. This means that the integral can be expressed as the expectation E of $f(x)$ over the density $p(x)$ as:

$$\int_a^b h(x)dx = \int_a^b f(x)g(x)dx = E_{p(x)}[f(x)] \tag{2.2}$$

[37]. Considering the strong law of large numbers, drawing a large number of samples from $p(x)$ yields

$$\int_a^b h(x)dx = E_{p(x)}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i). \tag{2.3}$$

This process is called Monte-Carlo integration. In terms of optimization the generated samples can directly be used to approximate the maximum of the objective function $f(x)$ after

$$\hat{x} = \text{argmax}_{x_i, i=1\ldots n} f(x_i) \tag{2.4}$$

The first approach which served as the basis of modern Monte-Carlo sampling techniques was proposed by Metropolis and Ulam [78] and Metropolis et al. [79]. While Monte Carlo methods were often more efficient than conventional numerical methods, they required sampling from high dimensional probability distributions which was still difficult and expensive in terms of computation time. Therefore Hastings [43] generalized the metropolis algorithm, by proposing asymmetric proposal distributions which increased the efficiency of the algorithm especially for bad initial states. Today this algorithm is known as Metropolis-Hastings algorithm and is one of the most significant MCMC methods and meta-heuristics. This is because all MCMC methods in use today are to some extend extensions or special applications of the Metropolis-Hastings algorithm [3].

Instead of purely randomized generation of samples the main strategy of MCMC methods is to explore important regions of the finite state space. This is realized by simulating a Markov Chain that samples new candidate states X' from a proposal distribution $q(X'|X_i)$ that is based on the current state X_i. Generated candidates X' are then either accepted or rejected as new states X_{n+1} based on the acceptance probability

$$\alpha = min\left[1, \frac{p(x')q(x_i|x')}{p(x_i)q(x'|x_i)}\right]$$

It has been proven that the Metropolis-Hastings algorithm asymptotically reaches a stationary distribution that is equal to $p(x)$ independent of the initial starting point of the chain [121]. For multimodal target distributions the choice of initial starting values becomes more important as the Markov Chain gets easily stuck in local modes. The most straightforward solution to overcome this problem is to initialize multiple

Markov Chains with different initial states.

Simulated Annealing

With respect to global optimization, instead of approximating some distribution $p(x)$, one is interested in finding the global optimum of $p(x)$. Hence, this requires formulating the problem as shown in Equation (2.4). A very popular MCMC-based approach in global optimization is *Simulated Annealing*. It adopts the idea of annealing in metallurgy, in which the crystal grain size is adjusted by heating material above its recrystallization temperature and subsequently cooling it at a certain rate. Computationally, the concept of temperature can be implemented as the probability of accepting uphill moves (transitions to states that are worse than the current). At the beginning of the algorithm the temperature is high, which results in a higher probability of moving towards a worse solution. This is important as it allows a sufficient exploration of the solution space and prevents the algorithm from becoming stuck in local minima. As the search continues, the temperature progressively decreases, resulting in less overall movement. The major difference between Simulated Annealing and the Metropolis-Hastings algorithm is the formulation of the acceptance probability, which is given by

$$\alpha_{SA} = min\left[1, \left(\frac{p(x')}{p(x_i)}\right)^{T(t)}\right] \tag{2.5}$$

where $T(t)$ denotes the cooling schedule [123]. Although the idea of Simulated Annealing is quite simple, Press et al. [97] point out that an effective implementation of the algorithm requires specifying a number of problem-dependent elements:

- Description of possible system configurations
- Generation of random changes in configurations
- Specification of an objective function that needs to be minimized
- A cooling schedule that encodes how the temperature T is lowered with respect to the iterations

3. Soft Foam Hands - Design and Fabrication

This chapter introduces the novel class of soft foam robots and specifically soft foam hands. The motivation for using foam as core material is founded in its lightweight nature and the large reversible deformations it can undergo. Additionally foam is low-cost, fast and easy to fabricate and available in various stiffnesses and densities. First experiments with using foam as a core for soft manipulators have been conducted prior to this work. While the initial prototype was designed as an anthropomorphic hand, the concept of using foam structures is generalized to several other non-anthropomorphic designs in joint work with Cornelia Schlagenhauf [108]. The long-term goal of this project is to provide others with the tools to design, fabricate and control these highly-customizable, dexterous and inherently safe class of robot hands. To achieve this goal, various research has been and is still concerned (see Chapter 6) with topics such as control, modeling and design. This thesis in particular focuses on the automated design of soft foam hands. In this chapter the current state of the art of foam hands is described. First, the design space is defined, followed by a brief introduction of how foam hands can be modeled and concluded by instructions and details on the fabrication process.

3.1. Hand Design

Traditional rigid-body robot hands display impressive capabilities. Newer models are highly dexterous, capable of a large number of grasps or manipulations and come with a large array of sensors (force, tactile, etc.). Still, such highly-developed robot hands can rarely be seen outside of research labs. Issues such as the uncertainty and complexity of real-world environments or the lack of robust grasping certainly explain these circumstances. However, it is mostly the cost of such hands and the complexity requiring expert knowledge to operate them, which prevents dexterous robot hands from becoming more widespread in use. Another key problem with traditional robotic hands is that they are not easily customizable for a desired task.

Therefore the primary motivation behind soft foam hands is to create a class of robots that circumvents the shortcomings of traditional rigid-body hands by being inexpensive, customizable and inherently safe. To ensure accessibility for non-experts, these hands rely on off-the-shelf materials in terms of support and actuation and can easily be fabricated. The design space of foam hands spans across several different domains. When designing a new hand the most important parts to consider are the actuation, hand morphology and rest pose. Up to this point three foam hands have been designed which are shown in Figure 3.1, with the initial design on the *left* and two designs developed during this thesis in joint work with Schlagenhauf [108] in the *center* and *right* image of Figure 3.1.

3.1.1. Actuation and Morphology

Since soft robots do not have any joints, the variety and complexity of achievable poses largely depends on the actuation which is defined by the tendon routing. A major weakness of the initial design shown in Figure 3.1 *left* is the inability of the thumb to abduct and oppose the palm. This is mainly caused by an inefficient tendon routing with two antagonistic tendons, as shown in the left column of Figure 3.2. Changing the routing increases the complexity of feasible motions of the thumb significantly, enabling either lateral or opposing grasps. This is visualized in Figure 3.2 *right*. Besides the actuation, experiments have shown that the rest pose of the hand design predefines the range of motion independently of the tendon arrangement. Since the shape of the foam is fixed and cannot be changed (unlike the tendons), being able to model and evaluate the deformation of the foam in simulation prior to fabricating the actual foam model is therefore crucial. Depending on the underlying task, certain poses are identified to be

Figure 3.1.: Soft foam hand robot prototypes. *Left*: Initial prototype, anthropomorphic hand in a 'cupped' rest pose. Middle: Anthropomorphic hand in a stretched out rest pose. Tendons only run on the inner side of the hand. *Right*: Non-anthropomorphic four fingered gripper.

more suitable than others. This especially applies to human-like hand geometries. Hands with such flat rest poses have a problem when grasping large objects such as a tennis ball. This is due to the inability of the fingers to curl around the object and oppose the palm. However, an advantage of flat rest poses over curled rest poses is that they don't need tendons that run on the back of the hand, because the geometry and the compliance of the foam itself restores the hand to it's original shape. This makes it possible to add more tendons to the front of the fingers increasing the overall dexterity of the design. A design with a flat rest pose is shown in Figure 3.1 *center*.

Depending on the task, robot hands are required to achieve certain types of grasps and motions. While the morphology and the rest pose of the hand kinematically constraint the workspace of the hand, the motions within the workspace are most importantly determined by the tendon arrangement. This highlights an important advantage of the tendon driven approach, compared to e.g. pneumatically actuated designs, because of the ability to easily change the kinematics any time by switching to another differently routed glove. Overall, the design process of soft hands is subject to a multitude of contradictory parameters. Without experience and several design iterations it is difficult and tedious to achieve useful designs. Hence, modeling and providing appropriate simulation tools for soft hands is very important and are addressed in the work of Schlagenhauf [108].

3.1.2. Modeling

Section 1.1.1 discussed some approaches to model soft or continuous robots. In order to model the deformation of foam under the contraction of tendons this work follows the existing approach of Bern et al. [6] who use a finite element model (as described in Section 2.4) to capture the deformation behavior of soft plush toys. Their representation of soft plushies has been adopted and further refined by Schlagenhauf [108] to model soft robot hands. The resulting simulation framework was used in this thesis to predict hand deformations.

Since the model representation of hands provides the means for evaluating the design and dictates how the optimization problem is formulated, the following paragraphs will detail the modeling approach. For further details see Bern et al. [6] and Schlagenhauf [108].

Each foam hand is modeled as a discrete set of nodes denoted as $\mathbf{X}$ for the undeformed robot and $\mathbf{x}$ as the statically stable deformed pose. The total deformation energy of the system is defined as:

$$E = E_{foam} + E_{tendons} + E_{pins} \tag{3.1}$$

where E_{foam} is the energy due to deformations of the simulation mesh, $E_{tendons}$ is the strain energy stored by the contractile elements, and E_{pins} models the behavior of stiff springs that connect a small number of simulation nodes to world anchors in order to eliminate rigid body modes. The forces acting on each

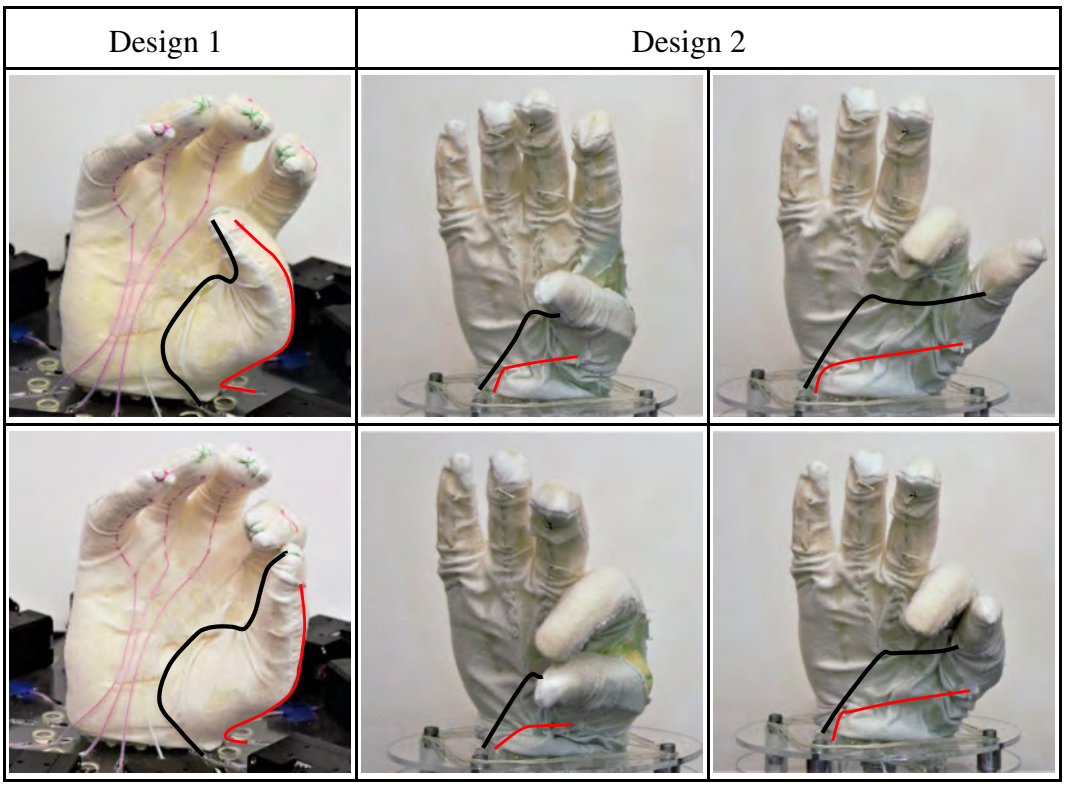


Figure 3.2.: Different tendon routings effect feasibility of motions. The antagonistic tendon routing on the initial prototype on the left enabled only a very limited range of motion. On the right, a changed tendon routing increases complexity of feasible motions, enabling opposing and lateral grasps. From [108].

node is given by the gradient of the energy functional with respect to the nodal degrees of freedom

$$\mathbf{F} = -\frac{\partial E}{\partial \mathbf{x}}. \tag{3.2}$$

The force Jacobian can be obtained through the Hessian of the energy

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = -\frac{\partial^2 E}{\partial^2 \mathbf{x}}. \tag{3.3}$$

Differentiation of the individual terms of Equation (3.1) thus yields the separated contribution of the deformation and the contraction to the total forces present :

$$\mathbf{F} = \mathbf{F}_{Foam} + \mathbf{F}_{contractile} + \mathbf{F}_{pins} \tag{3.4}$$

Foam

The elastic behavior of the foam is modeled using linear finite elements with a compressible Neo-Hookean material model. In order to derive element equations as described in step 2 of Section 2.4 the deformation gradient $\mathcal{F}$ is computed according to

$$\mathcal{F} = \frac{\partial \mathbf{x}^{\mathbf{e}}}{\partial \mathbf{X}^{\mathbf{e}}} \tag{3.5}$$

with e denoting the e-th element. Based on the compressible Neo-Hookean material model the energy density of each element is then given by

$$\Psi(\mathbf{x},\mathbf{X}) = \frac{\mu}{2}\mathrm{tr}\left(\mathcal{F}^{\mathrm{T}}\mathcal{F}-I\right) - \mu \ln J + \frac{\kappa}{2}(\ln J)^2 \tag{3.6}$$

with μ and κ being material constants, I the identity matrix and $J = \det(\mathcal{F})$. The total deformation energy E_{foam} is calculated by integrating Equation (3.6) over its constant domain and summing up the contribution of all elements together.

Tendons

Tendons are modeled as *contractile* elements that abstract the contraction of a tendon as changing the rest length of the underlying unilateral spring model. A contractile element is defined as a piecewise linear curve with two endpoints (x_s, x_t) and n intermediate vertices $(x_1, \ldots, x_n)$. An exemplary depiction of tendons routed along the mesh can be found in Figure 3.3. For the sake of simplicity it is assumed that all points of contractile elements are bound to nodes of the simulation mesh. The initial rest length l_0 of a tendon is defined by the sum of distances between the vertices as

$$l_0 = ||x_s - x_1|| + \sum_{i=1}^{n-1} ||x_i - x_{i+1}|| + ||x_n - x_t|| \tag{3.7}$$

The contraction level α_c of each tendon describes the contracted length as

$$l_c = l_0 \cdot (1 - \alpha_c) \tag{3.8}$$

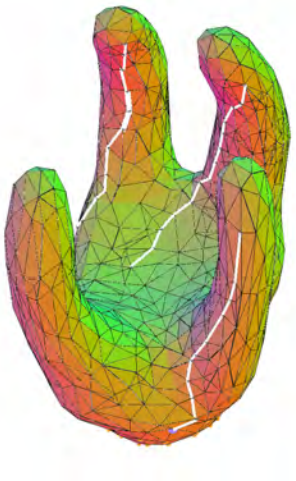


Figure 3.3.: Four fingered simulation mesh with three tendons routed on the fingers. Tendons run on the surface mesh and are highlighted in white color.

The unilateral strain energy U of each tendon is modeled as a piecewise C^2 polynomial $U(\Gamma)$ which models the strain with respect to the deformation Γ. More details on how the unilateral strain energy is modeled can be found in Section 4.3 of [6].

Pins

The pins restraint the foam in space. A pin is modeled as stiff zero-length spring. Any node in the simulated mesh can be specified as a pin by the user. With respect to the physical version of the foam robot they correspond to the fixation of the hand on the platform.

In the following chapters, the word *routing* refers to the choice of endpoints and intermediate vertices of each tendon. After the total energy equation of the system has been obtained the resulting deformation for a tendon routing with the contractions α_c can be calculated. This is done by using an direct sparse LDLT Cholesky solver to obtain the mininmum total energy state of the system.

Freeform simulation meshes can be created using *SculptGL*[1], a free online sculpting tool, and refined or remeshed with *MeshLab*[2].

[1] https://stephaneginier.com/sculptgl/
[2] http://www.meshlab.net/

3.2. Fabrication

After initial hand geometries, rest poses and tendon routings have been sufficiently explored in simulation and a purposeful design has been defined, the physical hand can be fabricated. The emphasis of the fabrication process is to ensure the use of low cost, readily available materials and keeping the overall process simple and quick. Depending on the individual experience and skills of the user fabricating a new foam hand from scratch is a matter of 1-3 days. While even most of the time is taken up by the curing process of the mold and the foam.

3.2.1. Creating the Mold

Depending on the hand design two options for creating a mold have been established. In case a human hand serves as model for the foam, alginate is used to get a negative of the hand which is then filled with plaster to create a replica of the hand. This process is depicted in Figure 3.4 (1) to (4). One side of the plaster hand is then covered with clay while the other side is covered with silicone (5). After curing, the clay is removed, mold release is applied and the remaining side of the hand is covered with silicone (6). Then the plaster is removed and the silicone mold is cleaned from residue (7).

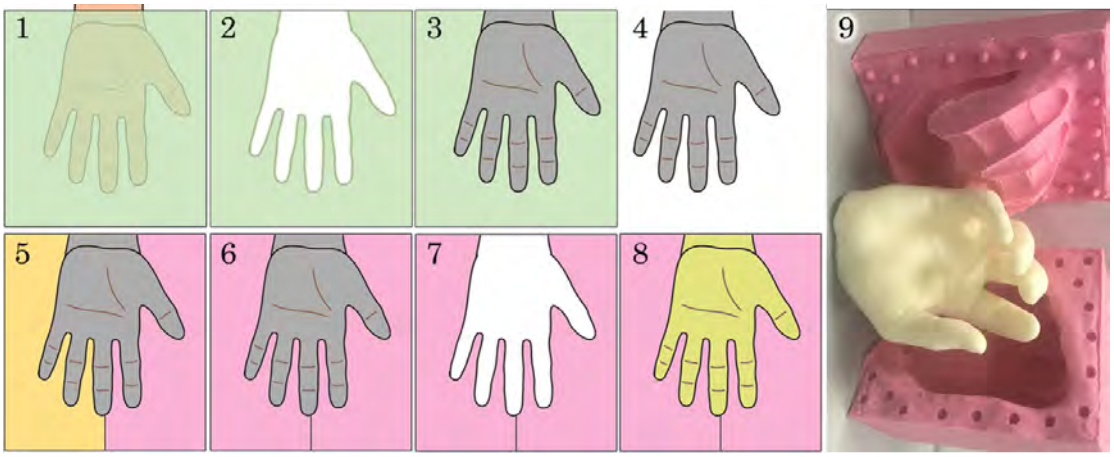


Figure 3.4.: Fabrication Process: 1) Cast of human hand in alginate, 2) Remove hand from alginate mold, 3) Pour plaster of paris into alginate mold and allow to cure, 4) Carefully remove plaster of paris cast from alginate mold, 5) Cover one half of plaster of paris hand with clay and the other half with silicone, 6) Remove clay, apply mold release agent to silcone and fill second half with silicone, 7) Remove the plaster of paris hand from two part silicone mold and clean the mold, 8) Use the master two-part silicone mold to cast foam hands, 9) Final result. *Image by J.King*

Another option for creating molds is to directly print a mold using standard resin or filament-based 3D printers [108]. Whenever non-anthropomorphic hands serve as basis for the mold, the hand geometry can either be turned directly into a mold manually using CAD-Software or using automatic mold-generation methods [130, 70]. An example for a 3D-printed mold is shown in Figure 3.5

Figure 3.5.: 3D printed mold and foam cast for a 2D planar gripper foam robot.

3.2.2. Casting Foam Hands

For casting hands a two-component polyurethane foam compound [3] is used. There a various types of foam densities available depending on the application. In terms of soft hands *FlexFoam-iT! X* performed best as it provided the best overall balance between stiffness and compliance. Examples of casted foam structure are shown in Figure 3.4 and Figure 3.5. A great advantage of this technique is, that the approximate cost for creating a 3d-printed reusable mold are around 50$ which enables the production of each further foam hand for a fraction of the cost. The ability to make a large supply of foam hands in a time and cost efficient manner is very important for lowering the entry barrier for researchers desiring to experiment with soft robots.

3.2.3. Gloves and Sewing Tendons

In order to place tendons on the foam a textile skin, e.g. a glove, serves as the connection between the foam core and the actuation. This can either be an off-the-shelf glove in the case of anthropomorphic hands or sewn skins for general robots. In the case of non-anthropomorphic designs a custom knit glove was used. These custom gloves were knit using the automated knitting process developed by [77].

Tendons are directly sewn into the textile skin using a needle and are fixed with a finishing knot. As material PTFE coated braided fishing line is used. To prevent the glove from slipping on the foam, a spray-on adhesive [4] is used to tightly bound both components together.

3.2.4. Robot Platform

Generally, soft foam hands could be mounted on any existing robot platform, either humanoid robot or robot arm. For first experiments and design validation a simple platform focusing on the hand itself was designed by Schlagenhauf [108]. This platform is easily manufactured and relies on standard do-it-yourself supplies as well as lasercut and 3D-printed parts. The motors [5] and the hand are mounted onto a stack of lasercut acrylic plates. Tendons are routed through PTFE tubes from the hand to the pulleys that connect to the motors. The foam hand is fixed on the top acrylic plate using hot-melt glue. This top plate can be easily switched out emphasizing a modular design that allows quick testing of different hand designs. The full setup including an exemplary anthropomorphic foam hand can be found in Section 3.2.4.

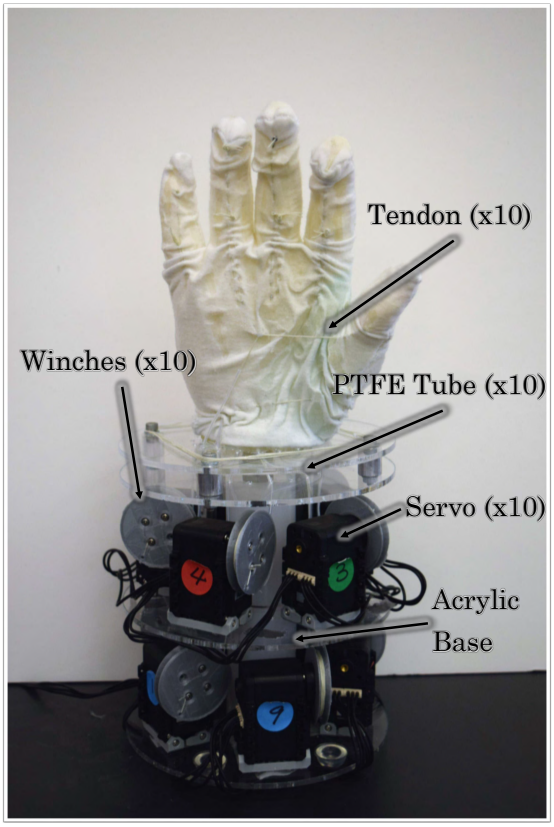


Figure 3.6.: Assembled platform of an anthropomorphic prototype with 10 Dynamixel motors mounted on acrylic plates. Tendons run through PTFE tubes and are connected to winches mounted on the motors.

[3] *Smooth-On FlexFoam-iT! Series*
[4] *3M Adhesive 23*
[5] *Robotis DYNAMIXEL AX-12A*

4. Automated Design of Tendon-Driven Soft Foam Hands

The absence of joints and links as well as the infinite amount of possible tendon routings creates a complex and unbounded design domain. For a robot hand to execute a desired task, the task can be represented as a sequence of grasps or poses, that the hand must be able to achieve sequentially. Finding the necessary tendon routings that are non-trivial is a challenging task, especially for non-experts. The proposed design pipeline provides a tool that enables users to automatically design task specific soft hands.

Overall this process is drafted to consist of the following steps:

- Specify and record desired grasps or manipulations.
- Automatically create a hand morphology and rest pose that is capable of reaching the desired poses.
- Generate tendon routings and respective contractions that achieve the specified goals for the fixed hand shape.
- Fabricate and build the actual soft hand.

This work is not aiming at creating a precise control and grasping strategy for soft hands but at developing an automated design process for soft hands that can achieve as many specific grasps and manipulations as possible. The particular optimization strategy is hereby independent of the design goal, meaning that other more efficient optimization techniques for this problem may exist. However, the algorithmic approach presented in this work has been particularly optimized for the problem at hand and produces promising results. Further extensions and improvements will be discussed in Section 6.3.

4.1. Collecting Goal Poses and Motions

In order to implicitly design complementing tendon routings and contractions around a fixed hand shape it is important to be able to easily specify an explicit functional goal for this otherwise unconstrained problem. Desired grasp configurations are therefore directly adopted from recorded human demonstrations (Figure 4.1).

Although this poses the challenge of mapping kinematics of human hands with joints and links to the continuous nature of foam hands, users can rely on their intuition when grasping with their own hand. Compared to creating synthesized data of grasps in simulation (or virtual reality) this process enables users to execute grasps under full haptic feedback of the object. In this work, a readily available CyberGlove[1] featuring 22 resistive bend sensors was used to record the corresponding hand configurations. However, the recently increased interest in inferring human hand poses from image sequences for virtual reality applications has led to the development of computer vision algorithms [116] that can directly derive joint configurations from single camera images. Due to time and resource constraints this has not yet been implemented but offers great opportunities for future works to simplify the process of recording goals.

Since the recorded joint configurations cannot directly be transfered to the foam hand mesh in simulation, the goal is to focus on optimizing for tendon routings and contractions that yield poses or motions

[1] http://www.cyberglovesystems.com/

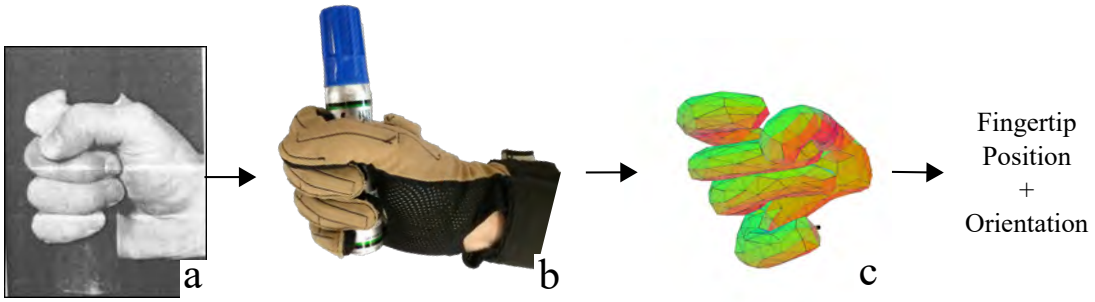


Figure 4.1.: A power grasp of a cylindrical object (a) (from [100]) is recorded using a CyberGlove (b). Fingertip positions and orientations are extracted and used to deform a hand simulation mesh into the desired goal pose (c). These positions and orientations then serve as targets for the optimization algorithm.

that closely reproduce the recorded fingertip trajectories. With the help of the CyberGlove SDK the 3D positions and surface normal orientations of the human fingertips are derived for each grasp. The combination of positions and normals then serve as goal trajectories for the optimization. This process is shown in Figure 4.1. Here, as an example, a power grasp of a cylindrical object is recorded and the target fingertip positions and orientations for the corresponding hand pose are obtained. The fingertip poses represent the goal for the proposed optimization algorithm.

In this context, a task can be represented as a sequence of hand poses, through which the hand transitions during the task. Hence, a manipulation task is obtained by recording poses sequentially.

4.2. Assisted Design of Compliant Hand Morphologies

Tendon actuated foam robot hands can produce motions that differ considerably from motions produced by rigid links and joints, and the reachable workspace of a foam hand cannot easily be inferred from its rest shape. While designing new soft hand models from scratch, certain types of grasps could often times not be achieved due to short or misplaced fingers or inefficient rest poses. For example, if one desires to build a soft hand that is able to pick up apples, the fingers of the hand should be long enough to properly enclose the apple to firmly grasp it. Ideally the foam should be able to assume a shape which enables similar contact points and configurations as they occur when humans grasp an apple. Based on the GRASP taxonomy by Feix et al. [31] a typical grasp humans could use for such a task is grasp 26 (Sphere 4 Finger, Power Grasp, Pad Opposition, Abducted Thumb). In order to design a foam geometry that is able to assume a kinematical equivalent pose when grasping an apple one needs to consider the position of the thumb, fingers, palm and their proportions. Without a proper reference that distinctively constraints the continuous design space of the foam it is almost impossible to produce purposeful designs. To provide aid with the design of task specific hand rest shapes, an assisted design process is proposed, in which a new hand mesh is obtained by "growing" along a set of 3D feature points. This way it is ensured that the designed mesh maintains the correct proportions independently of its rest pose and the user is able to successfully create unbiased hand meshes very quickly.

The feature points are obtained using a CyberGlove and the virtual human hand model provided by the CyberGlove SDK. For a given pose, 3D positions of the MCP, PIP and DIP joints and the distal end of the distal phalanges are recorded. An exemplary depiction of the resulting points is given in Figure 4.2 a).

Then a 3D grid is established in which an A* search algorithm is used to find the shortest path to connect the base of the palm with the set of recorded joint positions. Points on the 3D grid that are located

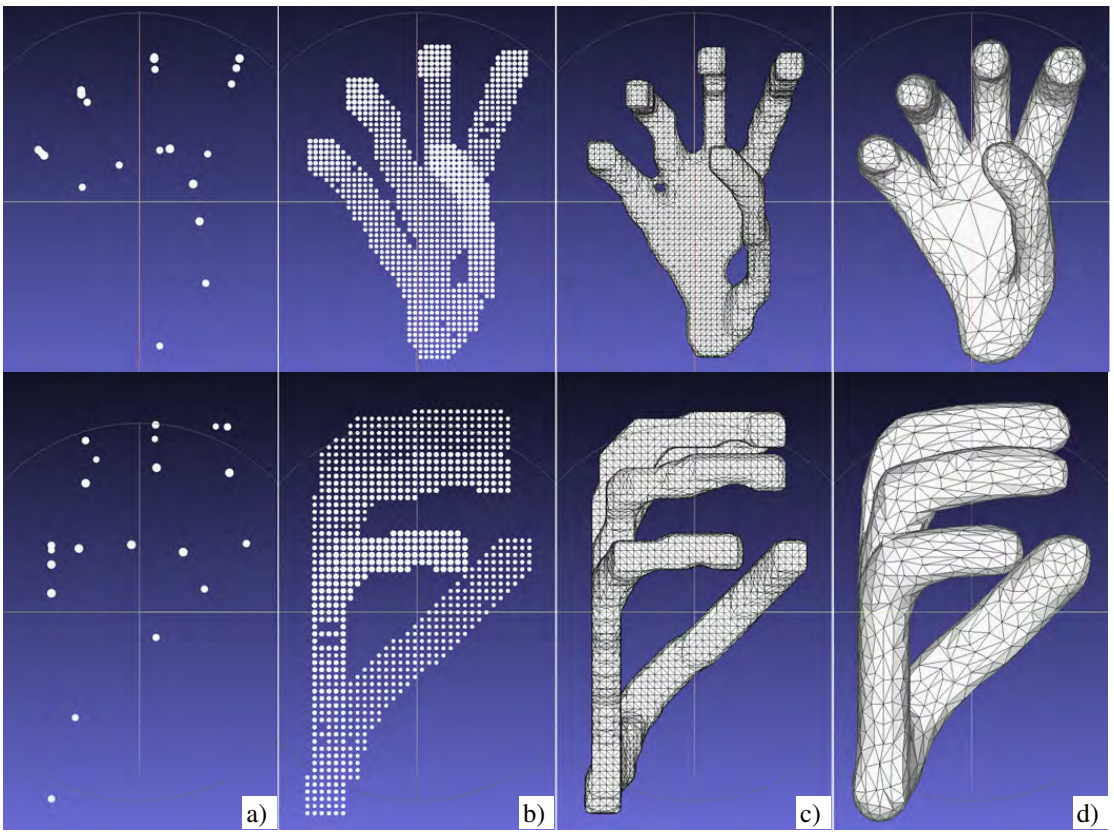


Figure 4.2.: "Growing" a new hand mesh along recorded joint positions. *a)*: 3D joint positions recorded using a CyberGlove. *b)*: Pointcloud obtained by path planning between the points. *c)*: A rough surface mesh created by surface reconstruction. *d)*: The resulting smoothed hand mesh.

along the shortest path are added to the set of joint positions. This set then contains a representation of points on the grid that are able to reach the desired grasp. In order to further increase the volume of the mesh, occupied points are then iteratively dilated towards directions of unoccupied grid points (Figure 4.2 b). The overall grid size and the amounts of dilation operations can be specified by the user. To create the meshes shown in this thesis a grid size of 200x200x200 is used and a total of two dilation operations were executed. Then the Delaunay tetrahedralization of the set of 3D points is calculated using *tetgen* [115]. The resulting convex hull is then shaped using an alpha shaping algorithm (Figure 4.2 c)). The final step consists of a series of mesh operations that are executed in *MeshLab* [18]. First the volumetric mesh is converted to a surface mesh. Finally, the overall surface is smoothed using an laplacian smoothing operation [119]. The resulting surface mesh is depicted in Figure 4.2 d). It is possible to refine or coarsen the mesh depending on the required accuracy of the simulation. This naturally is a trade-off between simulation speed and accuracy. Meshes that were created for the purpose of this thesis typically consisted of 1000 to 2000 nodes.

4.3. Optimization Algorithm

The optimization approach is built around the idea that a number of tendons is randomly placed on the hand mesh, contracted and the resulting pose being compared to the desired goal pose. As core algorithm a Metropolis-Hastings-Algorithm (Section 2.6.4) is used, which either accepts or rejects newly created tendon routings and activations based on their cost. To deal with the large amount of local minimums, additionally the concept of cooling a temperature as found in Simulated Annealing (Section 2.6.4) is

introduced. This allows the algorithm to accept solutions in the early stages that are worse than previous solutions and thus, escape local minima. The general process is best described by the pseudo-code found in Algorithm 1.

Algorithm 1 Tendon routing optimization

```
1:  T⃗,α⃗ = initialize (M)
2:  function SIMULATEDANNEALING(M,T⃗,α⃗)        ▷ M: Mesh, T⃗: Tendons, α⃗: Contraction levels
3:      CurrS = {M,T⃗,α⃗}
4:      while T > T_min do
5:          while i < MaxIterationPerEpoch do
6:              NewS = CreateNewS(CurrS)
7:              Cost = CalcCost(NewS)
8:              Acceptance = CalcAcceptance(CurrS, NewS, T)
9:              if Acceptance = true then
10:                 CurrS = NewS
11:             i++
12:         T = αT
13:         i = 0
```

While the temperature T is larger than a minimum temperature T_{min} the algorithm iteratively modifies the current solution, calculates its new cost and then either accepts or rejects the new solution. A solution is hereby encoded by the anchoring nodes along which the tendons are routed as well as the respective contraction levels in each timestep.

The individual steps are described in detail in the following.

4.3.1. Initialization

A tendon is represented by a number of waypoints along which it is routed, these points are termed *anchoring nodes* in the following. For each tendon the optimization is initialized by sampling n anchoring nodes from a discrete uniform distribution $\mathcal{U}(0,N)$ with N being the number of nodes in the mesh. Since in reality tendons are routed through the glove along the surface of the foam, tendons in simulation are restricted to run along edges on the surface of the mesh. Additionally, the first node of each tendon n_1 can only be sampled from fixed nodes at the base of the mesh. This is done to guarantee that on the physical robot, tendons begin close to the base of the hand where they can be connected to motors. After sampling the anchoring nodes, the corresponding tendon is obtained by running an A* algorithm to find the shortest path along the mesh that connects the sampled nodes.

4.3.2. Creating New Candidate Solutions

Creation of new solutions combines two separate sampling steps: One to create a new tendon routing $\vec{T}$, the other to sample new contraction levels $\vec{\alpha}$. To create new tendon routings, one anchoring node from each tendon is changed at each iteration. For switching out nodes, a heuristic is used which prefers transitions to adjacent nodes over transitions to nodes that are located further away.

In detail, this is realized by creating a set of neighboring nodes which are in direct or close adjacency to the node that is being changed. For this purpose a lookup table is constructed at initialization that contains the node IDs of adjacent nodes for each node in the mesh. This ensures that at each sampling step the d-adjacent nodes can be quickly determined without exhaustively searching the entire mesh.

The actual sampling step of a new anchoring node can be further divided into two individual samplings: First, the search depth $d \sim \mathcal{U}(0,D)$ is sampled from a discrete uniform distribution, with the maximum search depth D serving as a hyper-parameter that defines the maximum depth of nodes that should be considered for transitions. A depth of $d = 1$ hereby means, that only directly neighboring nodes are contained in the set of candidate nodes, whereas in the case of $d = 2$ also nodes that are adjacent to the

direct neighbors can serve as candidates for transitions. In the second step the actual node to which the solution transitions is uniformly sampled from the constructed set of neighboring nodes.

Due to the first uniform sampling step nodes that are closer are more likely to be contained in the set of candidate nodes and thus are more likely to be transitioned to after the second sampling step. This ensures transitions that do not significantly change the energy of the system, which is important to avoid random walks of the Metropolis-Hastings algorithm.

In addition to anchoring nodes, new contraction levels $\vec{\alpha}_{t+1}$ are created by sampling variations Δ from a normal distribution $\Delta \sim \mathcal{N}(0.0, 0.05)$ and applying them to the current contraction levels as follows:

$$\vec{\alpha}_{t+1} = \vec{\alpha}_t + \Delta$$

4.3.3. Evaluating Candidate Solutions

The main goal is to find tendon routings that can achieve certain grasps and transitions, thus cost of a tendon routing is primarily evaluated in terms of whether one or several goal poses are achieved. As described in Section 4.2, for each goal pose a set of fingertip coordinates $\{\vec{P}_1, \vec{P}_2, \ldots, \vec{P}_n\}_{goal}$ and normals $\{\vec{N}_1, \vec{N}_2, \ldots, \vec{N}_n\}_{goal}$ is recorded. Based on this goal specification, different variations of evaluating candidate solutions have been developed over the course of this thesis. The first approach calculates the cost of the solution as the RMS error of the euclidean distance between desired and current fingertip positions for a desired pose.

This measure is complemented by calculating the normals offset of each fingertip in radians using the law of cosines. The total cost of a solution is therefore a weighted sum of distance and normals offset and is calculated as described in Algorithm 2.

Algorithm 2 Calculate cost of tendon routing

1: **function** CALCCOST(*Solution*)
2: $totalCost = 0$
3: **for each** $p \in GoalPoses$ **do**
4: $cost_d(p) = \sqrt{\frac{\sum_{i=1}^{n} \left(\vec{P}(p)_{i,goal} - \vec{P}(p)_{i,current}\right)^2}{n}}$
5: $cost_o(p) = \left(\sum_{i=1}^{n} \cos^{-1}\left(\frac{\vec{N}(p)_{i,goal} \cdot \vec{N(p)}_{i,current}}{|\vec{N}(p)_{i,goal}||\vec{N}(p)_{i,current}|}\right)\right)$ ▷ n: Number of fingertips
6: $totalCost += \alpha_d cost_d(p) + \alpha_o cost_o(p)$
return $totalCost$

Depending on the problem, the weights α_d and α_o can be adjusted accordingly to account for different units or to weight one objective more important than the other. Since foam robots do not consist of rigid links and joints, and instead deform continuously, the joint angles of the human hand cannot be used to infer the similarity of a foam hand pose to a human hand pose. This is circumvented in the proposed cost function by using only the fingertip poses to evaluate the cost of a candidate solution.

4.3.4. Acceptance of Candidate Solutions

New tendon routings are accepted or rejected based on Algorithm 3 which is a typical implementation of an acceptance criterion for simulated annealing as already presented in Equation (2.5). If the new tendon routing performs better in terms of cost than the routing from the previous iteration, the new routing is always accepted. However, a greedy-search is avoided by also accepting uphill moves. The probability of accepting an uphill move hereby depends on the temperature T and the magnitude of the cost difference $\Delta_{cost} = Cost(NewS) - Cost(CurrS)$ according to

$$P_{acc}(T, \Delta_{cost}) = \mathrm{e}^{-\frac{\Delta_{cost}}{T}}.$$

The pseudo-code found in Algorithm 3 depicts the exact procedure for accepting and rejecting new solutions.

Algorithm 3 Acceptance of new solutions

```
1: function ACCEPTANCE(NewS, CurrS,T)
2:     if Cost(NewS) < Cost(CurrS) then
3:         CurrS = NewS
4:         if Cost(NewS) < Cost(BestS) then
5:             BestS = NewS                    ▷ Keep track of best solution
6:     else Cost(NewS) > Cost(CurrS)
7:         P_acc = e^(-Δcost/T)
8:         if P_acc > rand ∈ [0,1] then
9:             CurrS = NewS
```

This cooling schedule was chosen due to its simplicity and the sufficient convergence that was achieved with it (see Section 5.1).

4.4. Meta-Structure of Algorithm

As the algorithm needs to scale well to the high-dimensionality of this problem the optimization is run in multiple threads. Depending on available CPUs, a number of N threads is created. At startup each thread is initialized independently of another with a random tendon routing. Figure 4.3 depicts how the threads are arranged in a closed circuit consisting of neighborhoods (depicted as colored sections) in which communication between threads can take place. After a number of N_{epoch} iterations all threads report their best solution achieved so far and exchange it with better solutions from other threads that are located within the same neighborhood. The size of the neighborhood thereby dictates how quickly information about the optimization domain is spread.

The amount of iterations per epoch after which communication takes places and the size of neighborhoods serves as a hyper-parameter to guide exploration and exploitation. Larger neighborhoods and shorter epochs increase exploitation while small neighborhoods and long epochs improve exploration of the solution space.

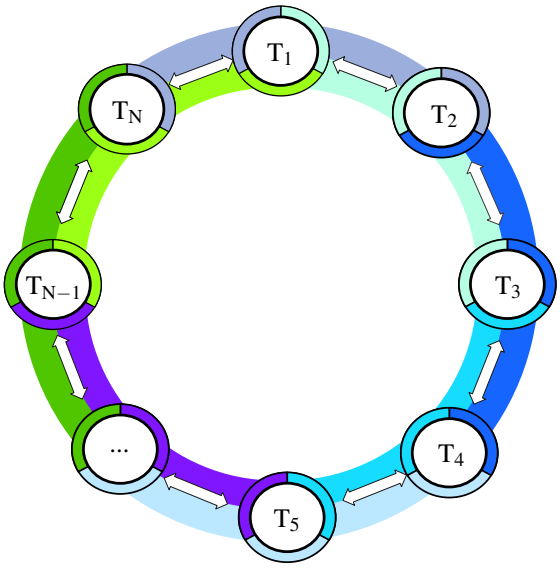


Figure 4.3.: Threads are arranged in a closed circuit consisting of neighborhoods depicted as colored sections in which threads can communicate and exchange solutions.

4.4.1. Cooling Schedule

Different cooling strategies such as adaptive cooling or thermodynamic cooling have been briefly tested. While these could further improve the convergence of the algorithm they have not been sufficiently tested and tuned to improve the performance of the optimization. However when solving for more complex problems in the future new cooling schedules should be investigated more in detail.

4.4.2. Parameters and Hyper-Parameters

This section quickly summarizes all hyper-parameters of the algorithm which require some sort of problem-specific tuning.

Number of tendons: The total number of tendons that should be used needs to be predefined.

Number of anchoring nodes: Defines amount of anchoring nodes along which tendons are placed.

Number of threads: Typically 2 threads per physically available CPU were used.

Initial temperature: The higher the temperature the more likely are uphill moves.

Terminating temperature: Temperature below which the algorithm terminates.

Cooling factor: Specifies how quickly the algorithm moves from exploration to exploitation.

Iterations per epoch: More iterations per epoch benefit exploration less iterations lead to exploitation of the solution space.

Standard deviation for sampling new activations: Larger deviations enable quicker convergence but bear the risk of changing the overall energy too drastically.

Neighborhood size: Defines how many threads can directly communicate with each other and thus how quickly information is propagated between threads.

Weight factors for distance and orientation: Define the amount and ratio of how much distance errors and orientation errors should be penalized.

All parameters mentioned are highly affected by differences in units ([mm],[cm],[m]), problem complexity and computational resources. Therefore, individual tuning of these parameters is always required.

4.4.3. Algorithm Termination

The optimization will not find the global optimum in an acceptable amount of time. To avoid excessive runtime, the algorithm terminates if either one of the following criteria are fulfilled:

- Cost of solution is below threshold
- Temperature is below minimum temperature
- No improvement of solutions after a certain number of iterations

4.5. Code Implementation

Besides determining the overall structure of the algorithm, a significant amount of work has been dedicated to improving the runtime and convergence of the optimization. Details on the actual implementation of the code are given below. The code base for simulating soft foam hands which existed prior to this work has been implemented in *Microsoft Visual Studio*. For optimization purposes and to enable the use of high-performance computing resources this code has been transferred to run platform independent on *Windows* and *Unix* based operating systems.

4.5.1. Languages and Libraries

The entire code is written in C++14 and utilizes features of the C++ Standard Library and the Eigen3.3.4 library for linear algebra. For multi-threading purposes the OpenMP 4.5 API is used. CMAKE (v.3.5.2) is used to support a cross platform and compiler-independent build process. The code is compiled using the GNU Compiler Collection (GCC v.6.3.0) utilizing all available vectorization features to improve computational speed.

4.5.2. High-Performance Computing

All results presented in this thesis have been obtained using the Extreme Science and Engineering Discovery Environment (XSEDE) which is supported by National Science Foundation grant number ACI-1548562 [122]. Specifically this work used the Bridges Cluster infrastructure at the Pittsburgh Supercomputing Center. Jobs have been submitted using *SLURM* batch scripts and typical optimization runs utilized less than 100 CPU hours.

5. Experiments and Results

This chapter first provides a number of experiments that verify the algorithmic concept and implementation. It then presents further experiments that explore the capabilities of foam hands in terms of their ability to achieve specific poses for a fixed morphology and an individual tendon routing. The results found through optimization are then evaluated in terms of their accuracy compared to designs created by humans. Finally, it is investigated whether the optimization is able to yield designs that can achieve sequences of poses. Therefore a number of transitions between poses are solved for and the results are presented at the end of this chapter.

5.1. Proof of Concept

In order to proof the functionality of the approach a number of experiments have been conducted which focus on verification of performance, precision, effects of parallelization and varying cost functions. For this purpose two meshes with different complexities are created and tested. The first mesh constitutes one simple finger while the other mesh depicts a full five-finger anthropomorphic hand created using the pipeline for assisted design introduced in Section 4.2. Given a goal pose or motion created by a known routing and contraction sequence the optimization should ideally yield the exact same routing as optimal solution. Whether this is the case is verified in the following.

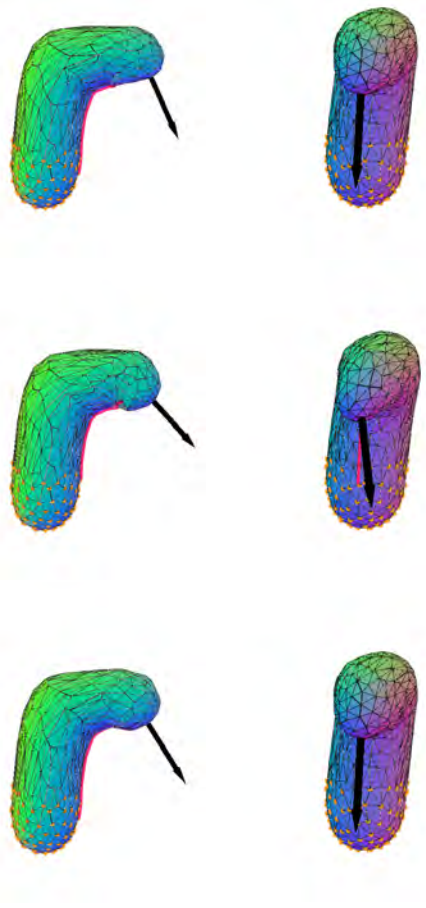


Figure 5.1.: A manually created pose resulting from one contracted tendon (top) is matched by the optimization. The fingertip normal is depicted as a black arrow. Middle: Test case 1: Cost is evaluated considering position. Bottom: Test case 2: Cost is evaluated considering position and orientation.

Verification of Algorithm Performance

To verify that the optimization is able to match simple poses, the following test is executed:
A pose is created by manual placement and contraction of tendons. This pose then serves as input goal for the optimization algorithm. The result of the optimization is then compared to the known routing and contraction.

Multiple tests of different complexities have been executed. For a first verification one tendon is placed on a single finger and actuated as depicted in Figure 5.1 *top)*. This pose serves as goal for two optimization runs with different objective functions. The first run (test case 1) calculates cost only based on the euclidean distance. As depicted in Figure 5.1 *middle)* the resulting pose closely matches the goal pose, however the orientation is not ideally matched. Adding the orientation as objective (test case 2) improves the overall

solution and results in the algorithm reproducing the exact solution, as shown in Figure 5.1 *bottom)*. This verifies that the general approach produces valid solutions in terms of routing and contractions and motivates further investigation of this problem in terms of performance. To subsequently verify different features of the algorithm and evaluate convergence, the pose from Figure 5.1 *top)* again serves as goal for the following test cases:

1. Optimization runs in single thread with one tendon and only euclidean distance as objective.
2. Optimization runs in single thread with one tendon, distance and orientation as objective.
3. Optimization runs in multiple (4) threads with one tendon, no exchange between threads, distance and orientation as objective.
4. Optimization runs in multiple (4) threads with one tendon, exchange between threads, distance and orientation as objective.

The different objective functions result in different absolute cost of solutions. Therefore, all convergence plots show the objective normalized with respect to the initial cost. The convergence of test cases 1 and 2 is compared in Figure 5.2 *left)* where the normalized objective is plotted against iterations. For this simple problem, the objective function has no significant influence on the rate of convergence. The comparison of test cases 2-4 in Figure 5.2 *right)* however shows a significant improvement of convergence for multi threading and exchange of solutions. Reasons for the acceleration of test case 3 (cyan) in comparison to the single thread case (blue) are funded in the random initialization which leads to some threads starting their search in more favorable regions than others. Test case 4 (red) converges even faster, as threads are able to exchange their solutions and thus sampling of new solutions is guided towards better regions more efficiently. Of course, the simple nature of the problem leads to extremely fast convergence overall and all solutions falling below the set threshold in terms of cost quickly. Thus, the results can only be considered qualitatively. Nevertheless, it captures the improvement of parallelization and communication of threads well and highlights their effects on exploration and exploitation of the parameter space.

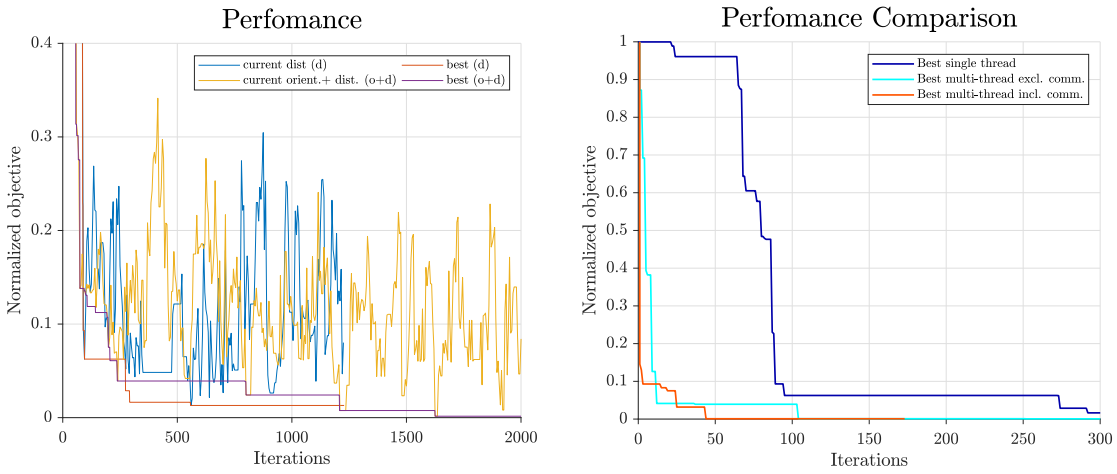


Figure 5.2.: *Left*: Convergence considering only the distance (d) and considering distance and orientation (o + d). The objective is normalized with respect to the initial cost. *Right*: Comparison of convergence for testcases 2 - 4, running the optimization with a single thread, multi-thread and multi-thread with exchange of solutions among threads.

The quantitative results of test case 1 - 4 are presented in Table 5.1. Since absolute cost values vary depending on which objective function is used, absolute cost values are normalized with respect to initial cost.

The results obtained for a simple mesh with one tendon justify the overall approach and demonstrate that the specific features of the algorithm fulfill their dedicated purpose. To verify whether the optimization also scales well to more complex poses, more complicated geometries and routings two further scenarios are tested.

Table 5.1.: Relative and absolute minimum cost of test cases [1-4] for a pose of a one finger simulation mesh and one contracted tendon. Costs are normalized with respect to the initial cost.

test case	normalized min. cost [%]	absolute min. cost []
1	2.29	1.327
2	1.30	0.754
3	0.062	0.036
4	0.086	0.048

Sequential Posing

The first scenario again investigates the simple one finger mesh but for a motion sequence created with two tendons routed along the side of the finger. The tendons are contracted in a sequence of three different levels as shown in Figure 5.3 *top)* and the resulting fingertip trajectory serves as goal. The solution found by the optimization is depicted in the bottom row of Figure 5.3. Although the routings are different, the resulting motion very closely matches the goal trajectory due to different contraction levels. The average position error of the fingertip is 0.17 cm and the average orientation error is 19.39 degrees. This verifies that the approach is also capable of optimizing with respect to pose sequences.

Figure 5.3.: Solution for a sequence represented by three different poses. *Top*: Target sequence. *Bottom*: Solution found by the optimization.

Anthropomorphic Hand Poses

An anthropomorphic mesh is created and tendons are placed on it in the same fashion as it is done in the simpler case with one single finger. Two test cases are created. The first is considered with reproducing

the pose created by one tendon while the second test case considers a more complex grasp created by three tendons. Figure 5.4 top) depicts relaxed and contracted target poses with one and three tendons. The respective best solutions found by the optimization is depicted in the bottom row of Figure 5.4. The relaxed poses are depicted for visualization purposes only and do not serves as optimization targets.

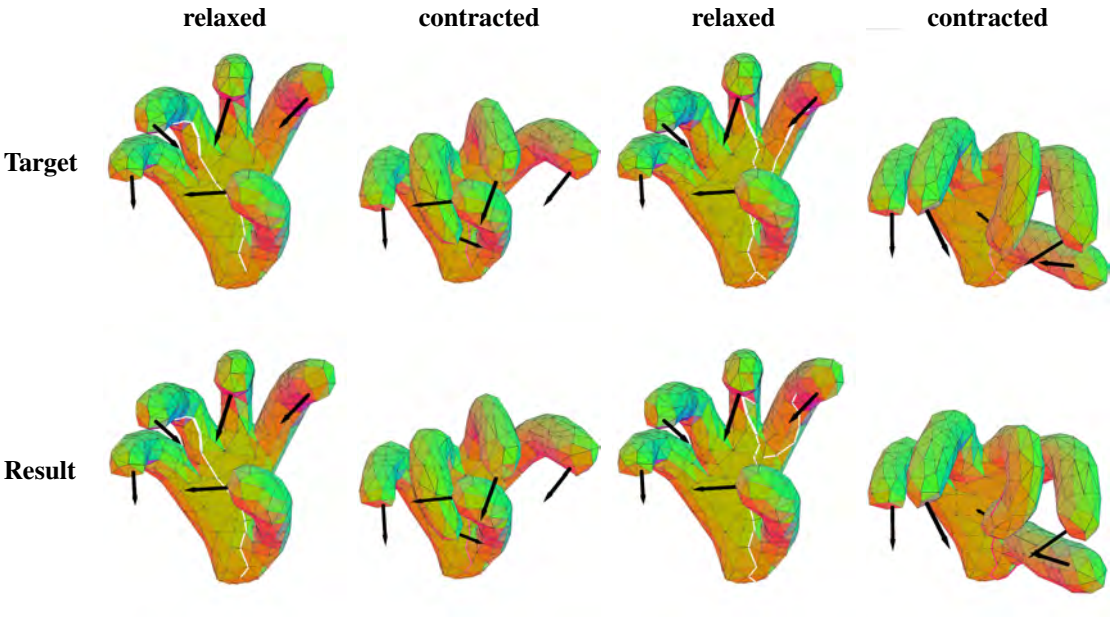


Figure 5.4.: Optimization results for two different poses of an anthropomorphic hand.
Top left: Relaxed and contracted target tendon. *Bottom left*: Optimization result.
Top right: Relaxed and contracted tendons for a target pose, using three tendons. *Bottom right*: Optimization result

For the target pose created by one tendon it is clearly visible that the result is identical to the tendon placement and contraction level of the goal pose. The only difference observed is the tendon being anchored to one node further below than the original routing. Since the nodes on the base of the mesh are fixed in space (*pinned*) this does not have any effect on the resulting deformation.

In the more complex case of three tendons the solution found by the optimization is not identical to the target. However the pose qualitatively and quantitatively matches the goal.

Overall, in both cases the optimization is able to find a routing that is able to produce poses that closely match the goal pose. In the case of one single tendon the solution is exactly reproduced while for three tendons the cost of the best solution is slightly higher.

5.2. Automated Design of Hands for Specific Grasps

The following experiments demonstrate and explore the kinematic capabilities of foam hands. The goal of this section is to quantify the variety of poses soft hands can achieve with individual tendon routings and how well the optimization yields such routings and contractions for a given morphology and rest shape. Investigating the ability of soft hands to execute single poses is especially interesting as this is the basic prerequisite for soft foam hands to execute a large number of poses sequentially and to thus carry out more delicate tasks. To verify that foam hands can indeed execute a large number of grasps, 14 grasps from different categories of the cumulative taxonomy (shown in Figure 2.4) are selected. The selected grasps are depicted in Figure 5.5 with the recorded grasp shown in the left images and the respective transferred pose in simulation on the right. It is important to note that for the purpose of visualization the simulated poses are all depicted using the same 5-finger anthropomorphic morphology with a cupped

rest pose. However, the target pose could be transferred to any other mesh with an arbitrary morphology and rest shape. The morphology depicted in Figure 5.6 is chosen to later compare the results to a user study (Section 5.3), which relied on a "human-like" morphology because most people can relate to 5-finger hands more intuitively. All following experiments have been executed using this mesh (Figure 5.6) unless stated otherwise.

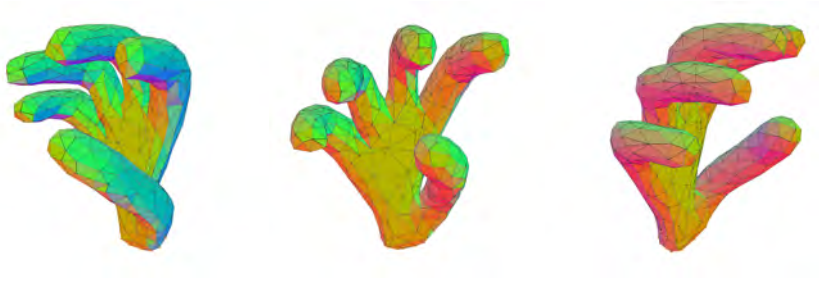


Figure 5.6.: 5-finger anthropomorphic hand morphology used in optimization.

Additionally a fixed number of 12 tendons and 3 respective 'anchors' per tendon are chosen for each optimization run. The three tendon anchors describe the start node, end node and one intermediate node on the surface of the simulation mesh, and a tendon is restricted to run through these anchors. The number of 12 tendons was chosen because it is considered to be a still realizable number in terms of motors needed and resulting complexity of the solution. More tendons would quickly result in very complex and difficult to manufacture routings. In case a solution is found that requires less tendons to realize a pose, some tendons in the solution are not contracted and can be omitted when transferred to the physical robot.

Table 5.2.: Optimization results for grasps.

Grasp Type	Mean Pos. Error [cm]	Pose
Power using palmar gutter **1.x**		
1.1a	0.895	yes
1.1b	0.934	yes
1.2	0.951	yes
Power using other parts of palm **2.x**		
2.1	0.989	yes
2.2	0.648	yes
Power with lateral stabilization **3.x**		
3.1	1.31	no
3.2	0.473	yes
3.3	0.529	yes
Precision with lateral stabilization **4.x**		
4.1	0.587	yes
Power with pad opposition **5.x**		
5.1	1.327	no
5.2	0.464	yes
Precision with pad opposition **6.x**		
6.1	1.399	no
6.2	0.518	yes
6.3	0.939	yes
Total	**0.855**	**11/14**

To quantify how many grasp types can be achieved within these constraints a solution is considered to be successful if the average position error of all fingers is $< 1cm$. This is motivated by the fact that state-of-the-art control approaches for fully compliant soft robots are typically suffering from an error of 10% (e.g. [103]) which at a typical finger length of $10-12cm$ equates to $\sim 1cm$. Furthermore, Schlagenhauf [108] has shown that the gap between simulation and reality for foam hands is within the same margins. In contrast to traditional rigid robots where this error margin would be disastrous, it has been demonstrated that due to their compliant nature, soft foam hands do not significantly suffer from such errors and are able to execute grasps successfully. Table 5.2 shows that a for a number of 11 grasps the pose was successfully matched while for 3 poses no tendon routing was found that sufficiently reproduces the desired grasp type.

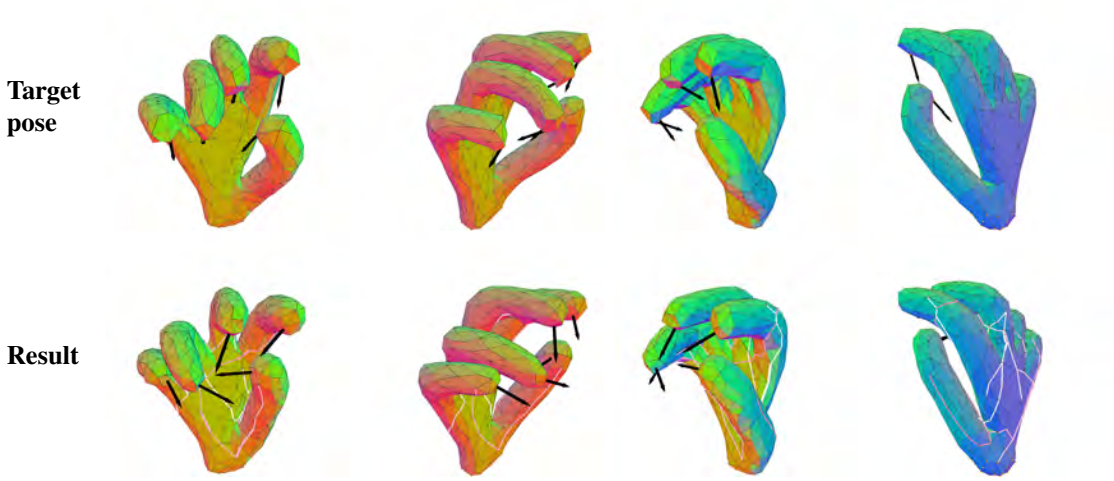


Figure 5.7.: Optimized tendon routing and contraction for a tripod grasp, shown from different camera perspectives. The result qualitatively matches the target pose closely.

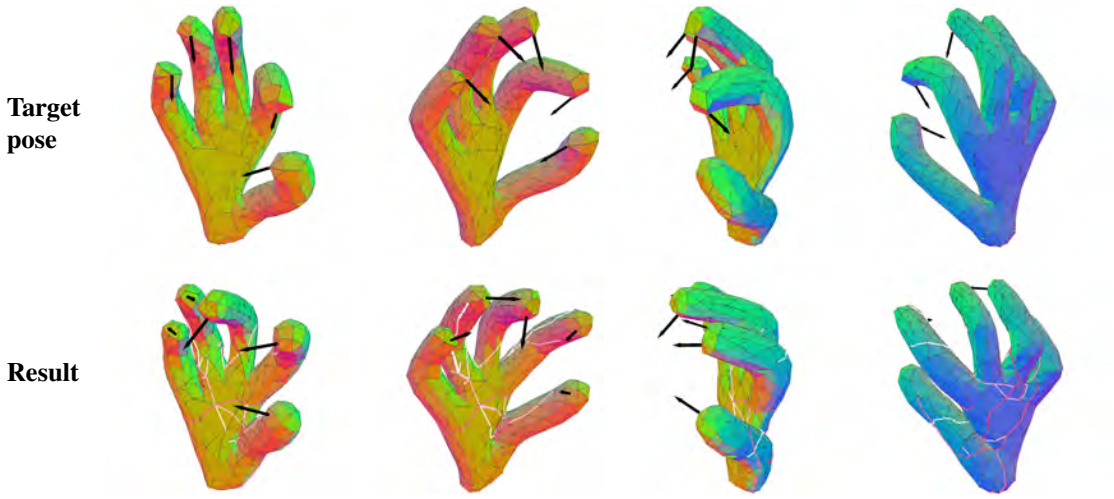


Figure 5.8.: Optimized tendon routing and contraction for a palmar pinch grasp, shown from different camera perspectives. No appropriate solution is found by the optimization, the resulting pose deviates significantly from the target pose.

Whether a solution is successfully found is hereby not determined by the class but depends on the individual type of grasp and the object size which was used to record the task. This is supported by the following two examples, which both depict solutions for precision grasps with pad opposition (class 6). Figure 5.7 depicts a tripod grasp (6.2) with an average position error of 0.518*cm*.

In contrast to this very good solution, Figure 5.8 shows a palmar pinch grasp (6.1) for which qualitatively no tendon routing and contraction is found. The quantitative error of 1.399*cm* further suggests that for the given morphology and rest pose no feasible solution exists. Possible reasons for this are funded in the recorded target pose itself. The size of the object used to record the grasp requires a significant abduction and simultaneous contraction of both index finger and thumb to form a precision grasp with pad opposition. While human hands are able to perform such a delicate task, the combination of tendons with foam does not allow such straightening without the foam being compressed by the tendon contraction. However, this problem could be circumvented by increasing the size of the foam hand or changing the morphology.

A complete overview of all solutions for the 14 selected grasps is visualized in Appendix A.1.

5.3. Comparison of Human Designs and Optimized Designs for Grasping

One proposition that justifies the optimization of tendon routings is that due to the high-dimensionality and the unintuitive deformation behavior of soft hands, it is difficult for human designers to create purposeful designs and to explore the full kinematic capabilities of soft foam hands. This section is dedicated to comparing the optimization results from Section 5.2 with human designs obtained in a user study. The user study has been executed by Schlagenhauf [108] and benchmarks the performance of tendon routings created by randomly selected people. In total 10 participants without a soft robotics background were asked to create tendon routings and contractions for three grasps (1.1a medium wrap a, 3.2 lateral tripod, 6.3 prismatic 3-finger). A time limit of 10 min per pose was set and users had enough time to familiarize themselves with the interactive design tools. Further details on the study and how the results were obtained can be found in [108].

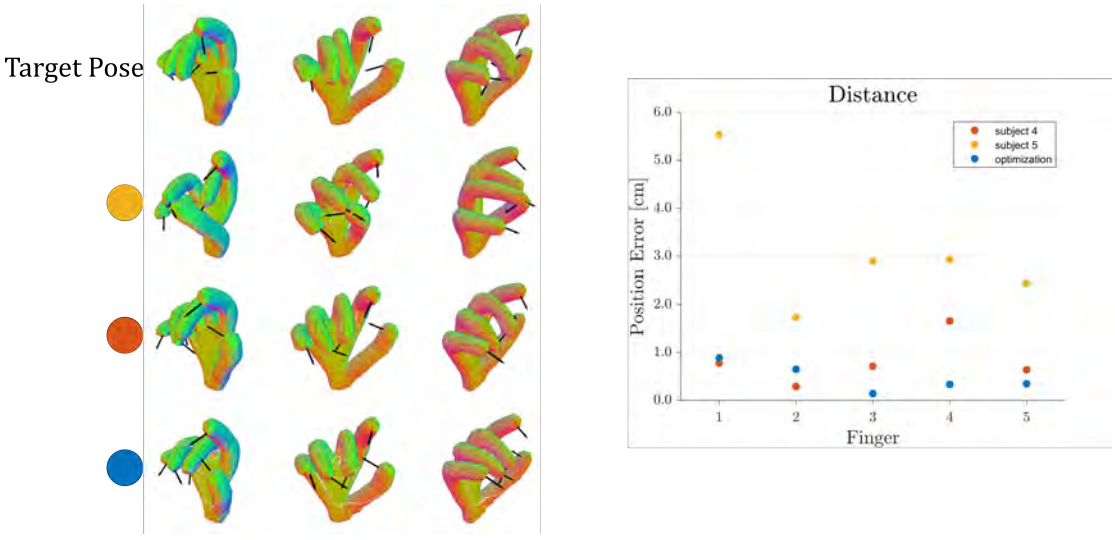


Figure 5.9.: Comparison of tendon routings created by two human study participants (orange, red) and the solution found by the optimization. *Left*: Hand poses. *Right*: Distance of the five fingers from the corresponding positions in the target pose.

Exemplary, Figure 5.9 compares the performance of two human designs (orange and red circles) and the optimization result (blue circle) for a lateral tripod grasp (target pose). The orange design does not closely match the desired pose qualitatively and the quantitative error depicted in the diagram on the right is >2cm. This is an example of a bad tendon routing designed by humans. An example of a very well designed tendon routing is depicted in red. This design qualitatively and quantitatively matches the desired pose very closely. In comparison to the optimization the red design is qualitatively equivalent in terms of matched pose and quantitatively even outperforms the optimization for the thumb (1) and index finger (2). Overall this shows that for a single grasp there are human designs that can compete with optimized tendon routings.

However, the average position error of the majority of human designs lies in between the red and the orange design and thus, cannot outperform the automated design. Considering all human designs for all grasps and comparing the errors of each finger quantitatively further supports this presumption.

The boxplot in Figure 5.10 shows that the median error of all human designs for all grasps is significantly larger than the error achieved by the optimization. Looking at the average individual error of each finger, the optimization is more precise than all human designs with a few exceptions in the case of index finger and middle finger. However, only a very small number of human designs was able to achieve slightly better results. Although the number of human test subjects in the study is relatively small and not representative of the entire population, the significant difference of average position error of the optimization (0.769cm) and human designs (1.379cm) strongly supports the assumption that for complex poses humans have difficulty to find precise tendon routings. This further supports the use of automated design techniques.

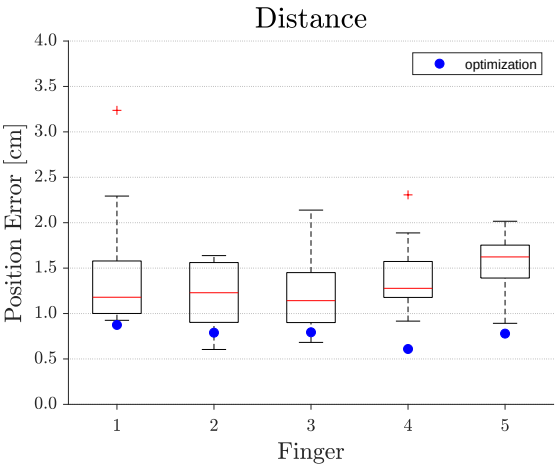


Figure 5.10.: Average Fingertip distances over all grasps tested in the study, compared to the average errors achieved by the optimization (blue dots).

5.4. Automated Design of Hands for Grasp Sequences

In the light of a growing task complexity robot hands that can achieve a large number of grasp transitions and can execute grasp sequences for in-hand manipulation are much-needed. This section presents several experiments that investigate whether the optimization is able to produce solutions for pose sequences. Instead of solving for one single target pose this means that the same tendon routing needs to be able to achieve multiple poses at different contraction levels. Therefore each target consists of several poses for which the cumulative average fingertip position error is evaluated. Due to this extension the intricacy of the problem is highly increased.

Task-Specific Designs

Most of the grasp sequences observed in humans can be expressed by transitions between classes of the cumulative taxonomy. However not all transition occur directly between grasp types. Often times transitions to intermediate grasps are required to successfully switch from one grasp to another. Therefore not all possible transitions are solved for in this section, but only a number of 4 transitions that could also be observed in human grasping are evaluated:

1. **Lateral** (3.3) → **Prismatic 3-finger** (6.3)
2. **Power Sphere** (5.2) → **Sphere** (3.1)
3. **Medium Wrap a** (1.1a) → **Small Diameter** (2.2)
4. **Tripod** (6.2) → **Lateral** (3.3)

The results of this experiment are depicted in Figure A.3 and the distance errors averaged over all fingertips are listed in Table 5.3. The average distance errors are larger than 1.15cm for all 4 grasp sequences. For each start and end pose, the error is notably higher than the average error compared to solving for the corresponding pose individually (compare to results in Table 5.2). This suggests that further testing and improvements are necessary to successfully create tendon routings and contractions for multiple hand poses. Potential reasons for high error values are identified and approaches to overcome them are suggested in the following:

- The optimization becomes stuck in local minima. Revisiting and tuning the algorithm to escape local minima could help overcome this limitation and should be addressed in future works.
- Tendons are restricted to run only along edges of the surface of the mesh, and the mesh resolution may be too coarse to create a routing capable of achieving several desired poses. Therefore future works should explore the use of finer meshes, this may however reduce the speed of the optimization process significantly.

Table 5.3.: Resulting distance errors for the 4 grasp sequence test cases. Distances are averaged over all five fingertips.

Average Distance Error [cm]	1)	2)	3)	4)
Start Pose	1.077	1.461	1.208	1.113
End Pose	2.210	1.657	1.279	1.204
Average	1.644	1.559	1.244	1.158

However, it should be noted that the error achieved by the optimization for 2 sequences is similar to the error achieved by an average human design for one pose (1.379cm). This emphasizes that the optimization produces promising results even for pose sequences.

General Purpose Designs

In addition to creating task specific soft foam hands that are able to execute specific grasp transitions, more general purpose designs are optimized for as well. For this purpose the ability of foam hands to achieve a large number of grasps with one single tendon routing is investigated. Instead of two subsequent poses, the optimization target consists of six poses, one pose of each class of the cumulative taxonomy. Since achieving 6 poses is expected to require a more complex tendon routing than achieving only 2 poses as described in the previous section, the number of tendons is increased to 15 for this experiment. The target consists of the following grasps (numbers corresponding to cumulative taxonomy):

Medium Wrap a (1.1a)	**Lateral Tripod** (3.2)	**Tripod** (6.2)
Small Diameter (2.2)	**Writing Tripod** (4.1)	**Power Sphere** (5.2)

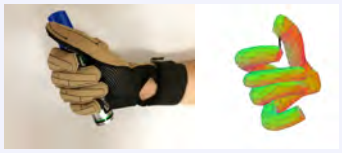
The resulting poses achieved by the optimization are depicted in Figure A.4, the distance errors for all poses are listed in Table 5.4. As already indicated by the results described in the previous section, the errors are larger than when optimizing tendon routing and contractions for the corresponding poses individually. As for the previous experiment, possible reasons could be founded in the algorithm becoming stuck in local minimums, and a possibly low mesh resolution.

Table 5.4.: Resulting distance errors for a generalized design, with one pose of each class of the cumulative taxonomy. Distances are averaged over all five fingertips.

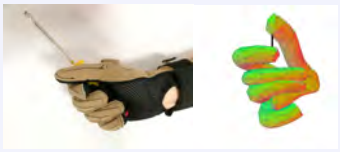
Grasp	**1.1a**	**2.2**	**3.2**	**4.1**	**5.2**	**6.2**	**Average Error**
Average Distance Error [cm]	2.643	2.357	1.291	1.214	1.752	1.233	1.748

The average error of $1.748cm$ achieved in this experiment is hereby only $\sim 27\%$ larger than the average human design error for one pose. This suggests that even though the optimization results for a large number of poses of different grasp types are suboptimal, the results are most likely still outperforming human design capabilities. Of course, this hypothesis needs to be verified by further extensive user studies, which evaluate human abilities to design tendons that can achieve several poses.

Power grasps using the palmar gutter



1.1a Medium Wrap a



1.1b Medium Wrap b

1.2 Ventral

Power grasps using other parts of the palm

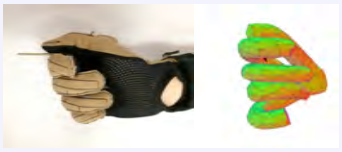
2.1 Palmar

2.2 Small Diameter

Power grasps with lateral stabilization

3.1 Sphere

3.2 Lateral Tripod



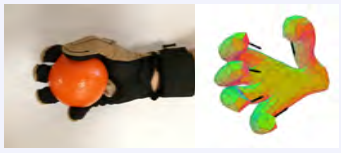
3.3 Lateral

Precision gasps with lateral stabilization

4.1 Writing Tripod

Power grasps with pad opposition

5.1 Large Diameter



5.2 Power Sphere

Precision grasps with pad opposition

6.1 Palmar Pinch

6.2 Tripod

6.3 Prismatic 3-finger

Figure 5.5.: 14 Target grasps are recorded with a CyberGlove and transferred to simulation poses. For each grasp, on the left, the target grasp is shown, and on the right the configuration of the deformed simulation mesh is depicted. Surface normals of fingertips are marked as black arrows.

6. Conclusion

This chapter concludes the work on automating the design of soft foam robot hands and summarizes the achievements and their implications. Suggestions for future work specifically considered with the further improvement of soft foam hands are also given.

6.1. Summary of Results and Contributions

Optimization of soft foam robot hands is a difficult multi-layered problem as designs are affected by morphology, rest shape, tendon routing and contractions simultaneously. In order to automate the design process of such hands a number of problems have been identified and addressed in this work. The main focus of this hereby lies on optimizing the tendon routing and contractions for a given soft hand morphology and rest shape.

In summary the research results and contributions that have been made are the following:

- Relevant design domains for soft foam hands that most importantly impact the ability of the design in terms of manipulation capabilities have been identified and specified.
- A universal, shape and rest pose invariant methodology for creating design goals for different hand poses that serve as input for optimizing tendon routings and contractions has been developed.
- Design tools were established to support the creation of multi-fingered hand geometries with different rest poses that are within kinematic constraints of the created goals.
- An optimization algorithm was designed that optimizes for efficient tendon routings and contraction levels for a fixed foam hand morphology (shape and rest pose).
- The algorithm was implemented to run on a high-performance computing cluster and demonstrated to successfully scale to different task complexities.
- The tendon routing optimization was proven to be able to produce meaningful routings for different task and shape complexities.
- A dataset of poses was created based on the cumulative taxonomy in order to evaluate the performance of the optimized tendon routings and to benchmark their capabilities in terms of variations between grasp types.
- Individually optimized routings for all poses from the dataset were presented. This was done for one specific anthropomorphic mesh with a cupped rest pose.
- The results of individually optimized poses were benchmarked against designs created by humans and quantitative results that optimized routings are more precise and efficient than manually designed routings were provided.
- The optimization was extended to solve for tasks by creating goals that contain sequences of multiple poses. Examples were given that the approach is able to produce solutions for such problems with certain limitations.

Overall, the results of this work show that in the future smart optimization techniques can indeed help to create hand designs that are more efficient and dexterous than currently available hand designs. Especially for the very complex multi-dimensional design domain of soft foam hands it has been demonstrated

that optimization helps to improve the performance of tendon routings and their respective contractions. Therefore, this work should serve as basis and inspiration for others to further improve and develop automated design processes to eventually enable the widespread use of soft foam hands that are purposefully designed for specific applications. With computational resources becoming increasingly available in the future, the approach of automating the design of manipulators and robots in general will become even more viable. The approach presented here should function as a starting point that can easily be extended to evaluate additional objectives or encode even more parameters such as material properties (density, Youngs modulus etc.), shape features or even dynamic object contact relations.

6.2. Current Limitations

While the achieved results are promising, soft foam hands are still a long way off from being deployed in applications such as pediatric care or fruit picking. Especially regarding the fully automated design of soft foam hands there are several shortcomings that are still inhibiting the full potential of this class of soft robots.

One limitation of the current approach is that only tendon routings and contraction levels are simultaneously optimized for while the morphology and the rest shape are not considered in the optimization. Additionally the optimization is only able to consistently yield precise routings for single poses, while for multiple poses the results are suboptimal for many grasp transitions. Reasons for this are likely found in the optimization goals themselves, which only contain fingertip positions and normals in discrete poses and do not consist of densely sampled trajectories. Another limitation is that self-intersection of the mesh is not detected which is problematic when optimizing for pose sequences because fingers can intersect in simulation during motions. A transfer of such routings and contractions to a physical robot would therefore likely yield motions in which fingers collide with each other. Apart from that, a transfer of solutions obtained in simulation to real robots has not been done yet. While the gap between simulated and real-world behavior for conventional routings was determined to be $\sim 0.6cm$ there is no concise way of transferring the complex and irregular routings found in optimization to the actual robot.

In terms of different hand morphologies, current limitations are implied by the creation of complex silicone or stereolithography molds. Especially undercuts due to irregular rest poses largely limit the manufacturing of suitable mold geometries.

With respect to the simulation the major issue is that it is not possible to simulate direct interactions with objects. Thus, optimization goals are limited to static poses and do not consider contacts or dynamic relations with objects.

6.3. Future Work

The current limitations of the automated design process presented here can be addressed and solved. A large amount of ideas had to be left untouched in the course of this thesis because of limited time available. Therefore, this section is a collection of ideas that have been developed together with Cornelia Schlagenhauf and can serve as possible starting point for future research.

Simulation

The optimization result is largely dependent on the quality and richness of the simulation. Currently, object interactions with hands are not simulated, which would be important in order to extract contact point relations and forces. Moving away from the static representation of poses towards simulating actual dynamic motions including a functioning contact simulation would enable to optimize for object-centric grasp representations. Most of the suggested improvements are based on a functioning contact simulation which makes this the top priority for future works. In her future works section Schlagenhauf [108] presents the current state of the contact simulation, explains which features have been implemented and provides an insight on the efforts taken so far.

Goal Creation

Under the assumption of having a functioning contact simulation available, various options for creating different optimization targets should be considered. Instead of grasping objects and recording the respective pose using a CyberGlove, optimization goals could be directly recorded in simulation similar to a virtual reality setting. For this purpose, Schlagenhauf [108] provides an interface between CyberGlove and simulation that maps recorded sensor positions to the simulation mesh. This would then allow to manipulate objects in simulation with any arbitrary morphology and simultaneously record contact point and object trajectories and contact forces. This information could then be used as goal for optimizing for tendon routings instead.

Transfer of Routings to Real Robots

So far the transfer of routings from simulation to the real robot has been done qualitatively. The achieved results are promising and the average position error of $\sim 0.6cm$ shows that the gap between simulation and reality is within the range of expected accuracy for a system as compliant as foam. However, more complex tendon routings resulting from the optimization will make it more difficult to qualitatively route tendons. To circumvent this problem two ideas are presented:

- Gloves for multi-fingered non-anthropomorphic hands have already been successfully knitted using the automated knitting algorithm developed by McCann et al. [77]. In addition to being able to automatically knit textiles directly from a 3d mesh input, their approach is also able to embed differently colored threads precisely where specified. This opens up the chance to directly knit differently colored markers into the glove along which tendons should be routed.
- Instead of using a glove along which tendons are routed, it would also be possible to embed small hooks directly into the foam which then serve as anchors for the tendon. Together with rapid prototyping techniques which could precisely place the hooks at the specified locations, this would simplify the overall manufacturing process.

Furthermore, friction between tendons and glove is currently not considered in simulation. Transferring routed tendons to the real robot will likely cause problems with friction depending on the complexity of the routing. To prevent this, a regularizer should be added to the optimization which penalizes tendons that are too long and erratic.

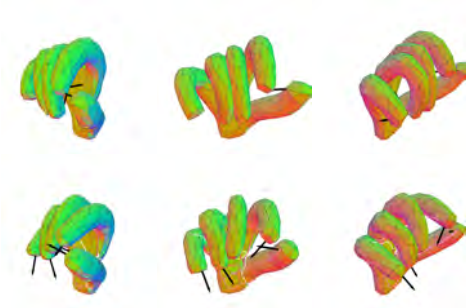
Creation of Real Foam Hands from Complex Morphologies

The tools created in this thesis which support the design of multi-fingered non-anthropomorphic hands enable the creation of very irregular soft foam hand morphologies. Depending on the rest pose, such morphologies can be even more difficult to design molds for. Automated mold generation algorithms [70] could solve this problem. However, more interesting solutions involve printing the foam hands directly using highly porous filaments that consist of a polymer and PVA compound. After printing, the material can be submerged in water to dissolve the PVA resulting in a highly compliant material behavior. Future works could evaluate whether such an approach is feasible as it could simplify the manufacturing process significantly.

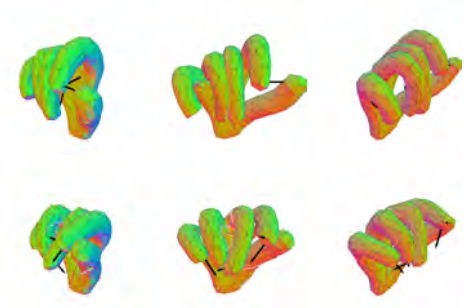
A. Appendix

A.1. Optimization results for grasps

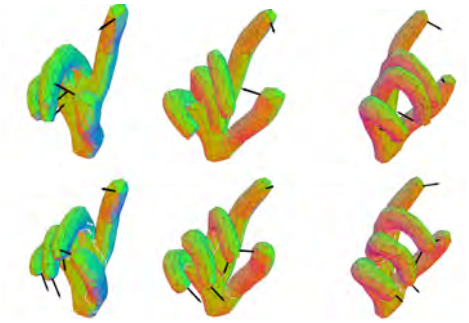
1.1a Medium Wrap a



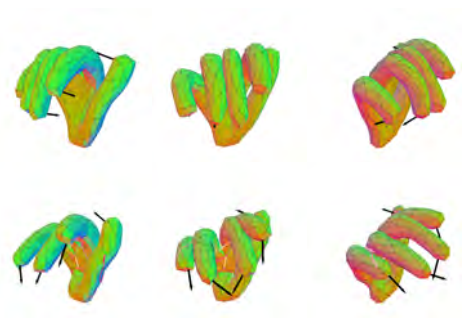
1.1b Medium Wrap b



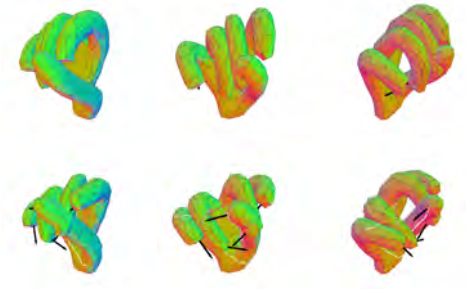
1.2 Ventral



2.1 Palmar



2.2 Small Diameter



3.1 Sphere

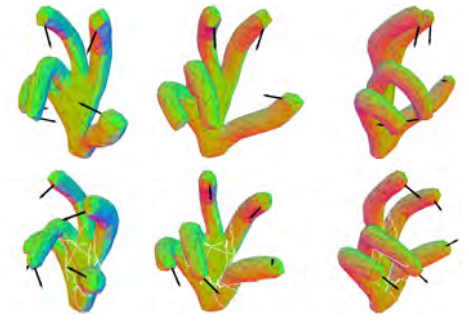
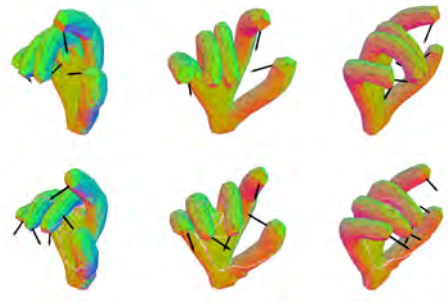
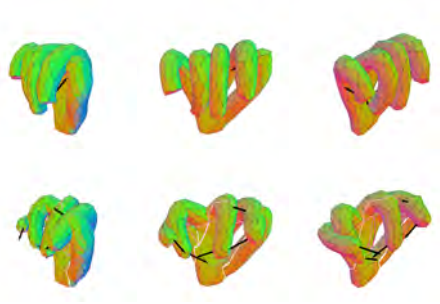


Figure A.1.: Optimization results for the 14 selected grasps (1). Top: Target pose. Bottom: Resulting pose.

3.2 Lateral Tripod

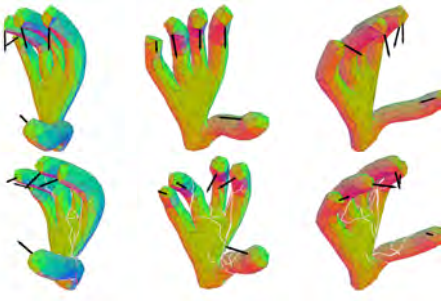


3.3 Lateral



4.1 Writing Tripod

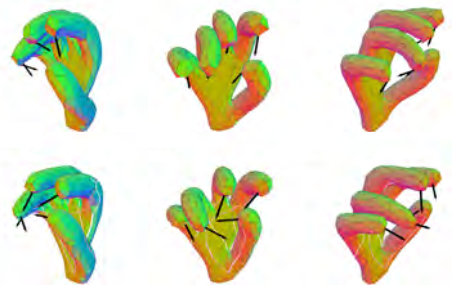
5.1 Large Diameter



5.2 Power Sphere

6.1 Palmar Pinch

6.2 Tripod



6.3 Prismatic 3-finger

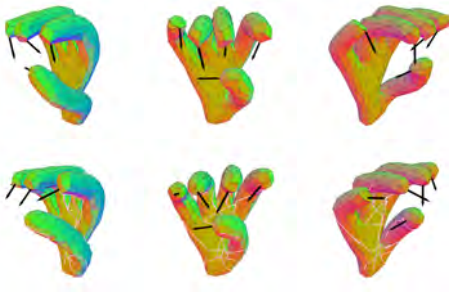


Figure A.2.: Optimization results for the 14 selected grasps (2). Top: Target pose. Bottom: Resulting pose.

A.2. Optimization results for grasp sequences

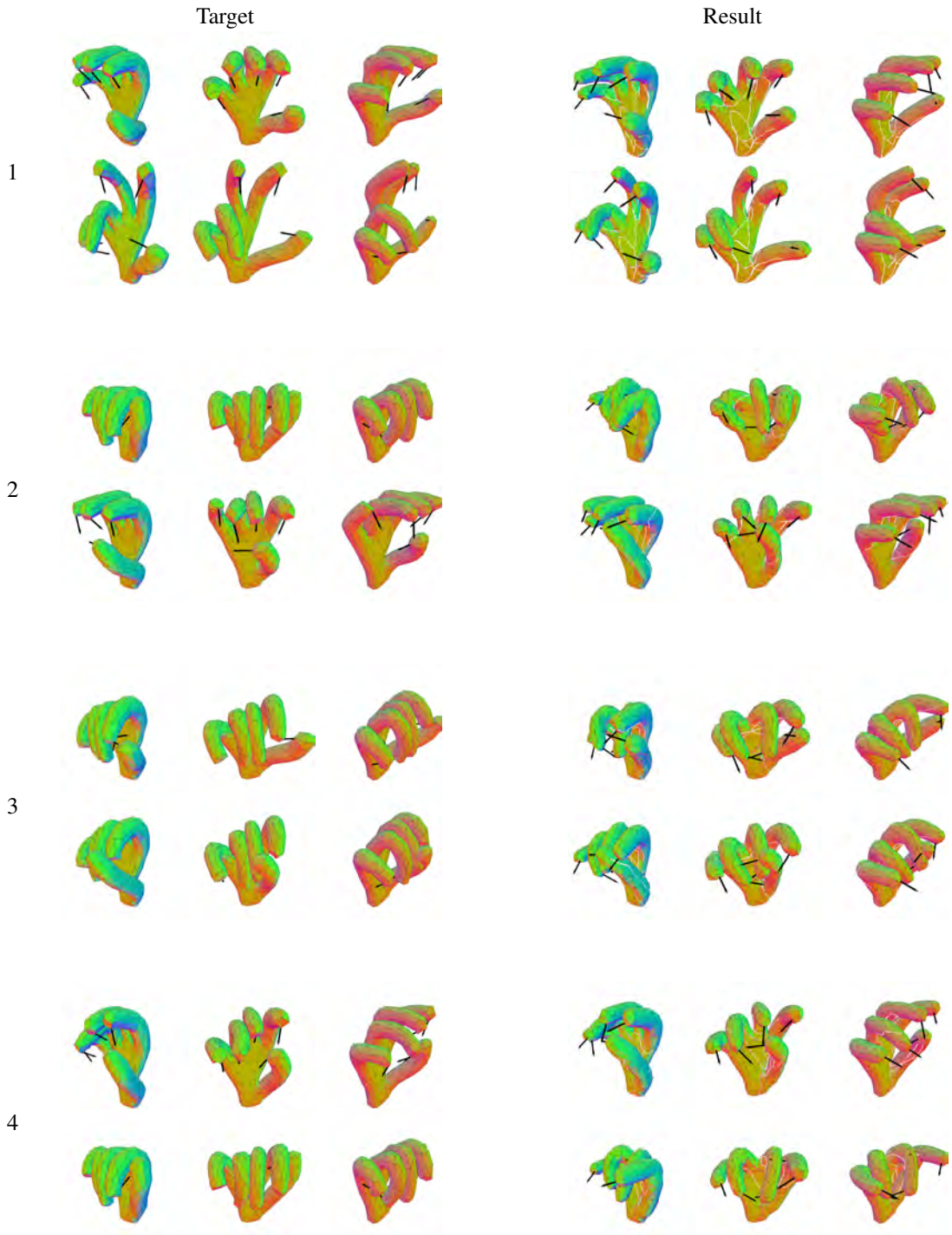


Figure A.3.: Results for grasp sequences. Left: Target sequence. Right: Resulting sequence.
1: Lateral (3.3) to Prismatic 3-finger (6.3)
2: Power Sphere (5.2) to Sphere (3.1)
3: Medium Wrap a (1.1a) to Small Diameter (2.2)
4: Tripod (6.2) to Lateral (3.3).

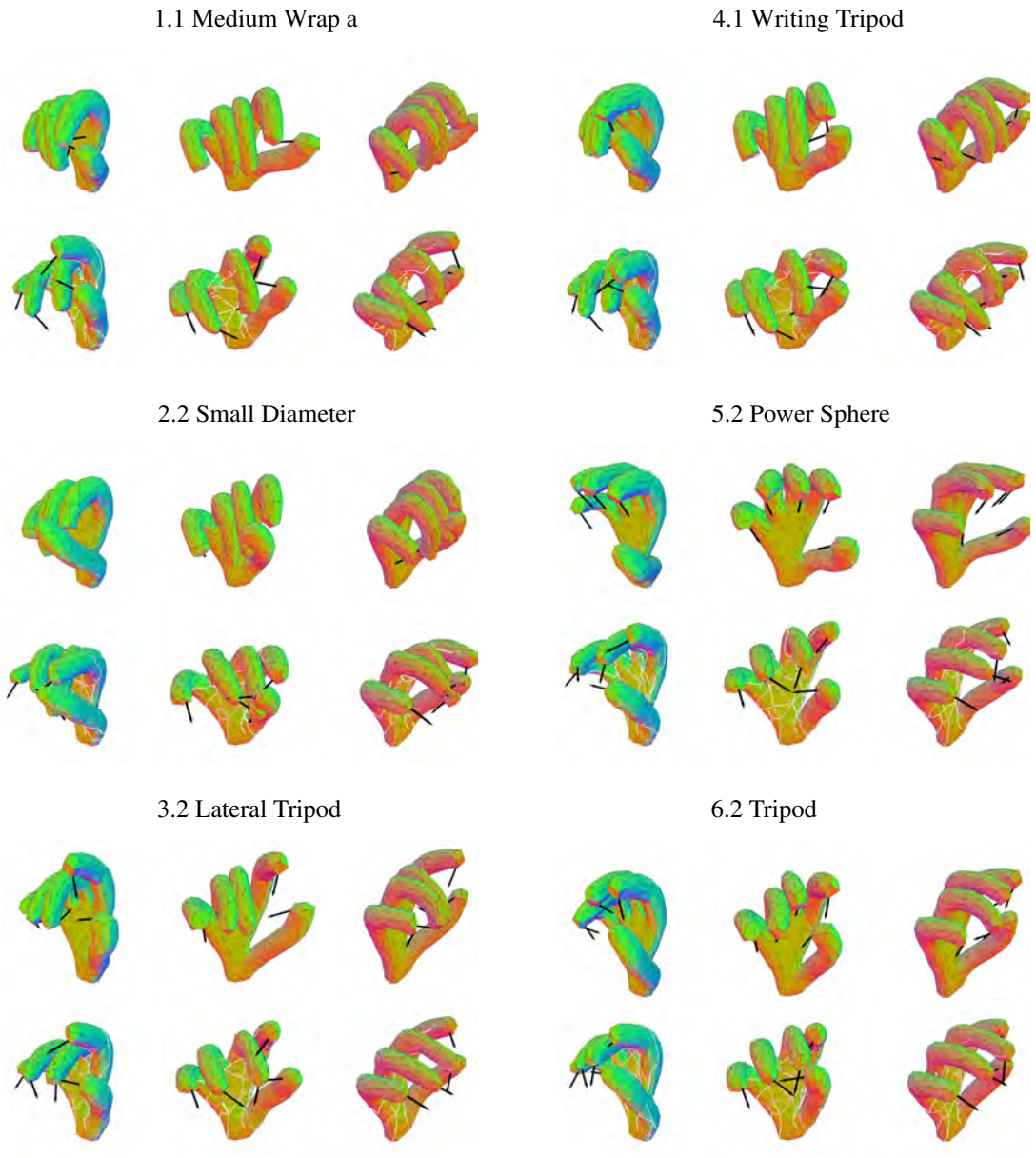


Figure A.4.: Optimization result for a general purpose design, optimizing one tendon routing to achieve 6 grasps, including one grasp of every group of the cumulative taxonomy. For every grasp, the top row shows the target pose, and the bottom row shows the achieved pose.

Bibliography

[1] S. Aaronson, G. Kuperberg, and C. Granade. Complexity zoo, 2005-18. URL https://complexityzoo.uwaterloo.ca/Complexity_Zoo. Accessed: August 13, 2018. 17

[2] M. Amjadi, K.-U. Kyung, I. Park, and M. Sitti. Stretchable, skin-mountable, and wearable strain sensors and their potential applications: a review. *Advanced Functional Materials*, 26(11):1678–1698, 2016. 4

[3] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC. *Machine Learning*, 50(1):5–43, Jan 2003. ISSN 1573-0565. doi: 10.1023/A:1020281327116. URL https://doi.org/10.1023/A:1020281327116. 21

[4] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. Armar-iii: An integrated humanoid platform for sensory-motor control. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 169–175, Dec 2006. doi: 10.1109/ICHR.2006.321380. 16

[5] P. Bachmann. Analytische zahlentheorie. In *Encyklopädie der Mathematischen Wissenschaften mit Einschluss ihrer Anwendungen*, pages 636–674. Springer, 1904. 17

[6] J. M. Bern, K.-H. Chang, and S. Coros. Interactive design of animated plushies. *ACM Transactions on Graphics (TOG)*, 36(4):80, 2017. 4, 15, 24, 26

[7] J. M. Bern, G. Kumagai, and S. Coros. Fabrication, modeling, and control of plush robots. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2017. 4, 5

[8] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, and H. M. Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010. ISSN 0027-8424. doi: 10.1073/pnas.1003250107. URL http://www.pnas.org/content/107/44/18809. 4

[9] I. M. Bullock, R. R. Ma, and A. M. Dollar. A hand-centric classification of human and robot dexterous manipulation. *IEEE Transactions on Haptics*, 6(2):129–144, April 2013. ISSN 1939-1412. doi: 10.1109/TOH.2012.53. 14

[10] J. Butterfass, G. Hirzinger, S. Knoch, and H. Liu. Dlr's multisensory articulated hand. i. hard-and software architecture. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2081–2086. IEEE, 1998. 15

[11] M. Calisti, M. Giorelli, G. Levy, B. Mazzolai, B. Hochner, C. Laschi, and P. Dario. An octopus-bioinspired solution to movement and manipulation for soft robots. *Bioinspiration & biomimetics*, 6(3):036002, 2011. 5

[12] L. Y. Chang and N. S. Pollard. Video survey of pre-grasp interactions in natural hand activities. 2009. 14

[13] W.-D. Chang and S.-P. Shih. Pid controller design of nonlinear systems using an improved particle swarm optimization approach. *Communications in Nonlinear Science and Numerical Simulation*, 15(11):3632–3639, 2010. 20

[14] E. Chao et al. Biomechanical analysis of static forces in the thumb during hand function. *The Journal of bone and joint surgery. American volume*, 59(1):27–36, 1977. 12

[15] F. J. Chen, S. Dirven, W. L. Xu, and X. N. Li. Soft actuator mimicking human esophageal peristalsis for a swallowing robot. *IEEE/ASME Transactions on Mechatronics*, 19(4):1300–1308, Aug 2014. ISSN 1083-4435. doi: 10.1109/TMECH.2013.2280119. 6

[16] I. Chiha, N. Liouane, and P. Borne. Tuning pid controller using multiobjective ant colony optimization. *Applied Computational Intelligence and Soft Computing*, 2012:11, 2012. 20

[17] M. Cianchetti, M. Calisti, L. Margheri, M. Kuba, and C. Laschi. Bioinspired locomotion and grasping in water: the soft eight-arm octopus robot. *Bioinspiration & biomimetics*, 10(3):035003, 2015. 5

[18] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In V. Scarano, R. D. Chiara, and U. Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. ISBN 978-3-905673-68-5. doi: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136. 31

[19] S. Cobos, M. Ferre, M. A. S. Uran, J. Ortego, and C. Pena. Efficient human hand kinematics for manipulation tasks. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2246–2251, Sept 2008. doi: 10.1109/IROS.2008.4651053. 10

[20] M. R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, June 1989. ISSN 1042-296X. doi: 10.1109/70.34763. 10

[21] M. R. Cutkosky. Climbing with adhesion: from bioinspiration to biounderstanding. *Interface Focus*, 5(4):20150015, 2015. 4

[22] R. Deimel and O. Brock. A compliant hand based on a novel pneumatic actuator. In *2013 IEEE International Conference on Robotics and Automation*, pages 2047–2053, May 2013. doi: 10.1109/ICRA.2013.6630851. 5

[23] R. Deimel and O. Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35(1-3):161–185, 2016. 5, 6

[24] R. Deimel, P. Irmisch, V. Wall, and O. Brock. Automated co-design of soft hand morphology and control strategy for grasping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1213–1218, Sept 2017. doi: 10.1109/IROS.2017.8202294. 6, 7

[25] C. Duriez. Control of elastic soft robots based on real-time finite element method. In *2013 IEEE International Conference on Robotics and Automation*, pages 3982–3987, May 2013. doi: 10.1109/ICRA.2013.6631138. 6

[26] C. Duriez. Control of elastic soft robots based on real-time finite element method. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3982–3987. IEEE, 2013. 14

[27] S. Edwards, D. Buckland, and J. McCoy-Powlen. Developmental & functional hand grasps. *Slack Incorporated, Thorofare, New Jersey*, 2002. 12

[28] S. Ekvall and D. Kragic. Grasp recognition for programming by demonstration. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 748–753, April 2005. doi: 10.1109/ROBOT.2005.1570207. 10

[29] J. M. Elliott and K. J. Connolly. A classification of manipulative hand movements. *Developmental Medicine & Child Neurology*, 26(3):283–296, 1984. doi: 10.1111/j.1469-8749.1984.tb04445.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8749.1984.tb04445.x`. 14

[30] J. M. Elliott and K. J. Connolly. A classification of manipulative hand movements. *Developmental Medicine & Child Neurology*, 26(3):283–296, 1984. doi: 10.1111/j.1469-8749.1984.tb04445.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8749.1984.tb04445.x. 12

[31] T. Feix, J. Romero, H. Schmiedmayer, A. Dollar, and D. Kragic. The grasp taxonomy of human grasp types. In *Human-Machine Systems, IEEE Transactions on*, 2015. 5, 10, 11, 14, 30

[32] N. Fukaya, S. Toyama, T. Asfour, and R. Dillmann. Design of the tuat/karlsruhe humanoid hand. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 3, pages 1754–1759 vol.3, Oct 2000. doi: 10.1109/IROS.2000.895225. 16

[33] N. Fukaya, T. Asfour, R. Dillmann, and S. Toyama. Development of a five-finger dexterous hand without feedback control: The tuat/karlsruhe humanoid hand. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4533–4540, Nov 2013. doi: 10.1109/IROS.2013.6697008. 16

[34] J. Gafford, Y. Ding, A. Harris, T. McKenna, P. Polygerinos, D. Holland, A. Moser, and C. Walsh. Shape deposition manufacturing of a soft, atraumatic, deployable surgical grasper. *Journal of Medical Devices*, 8(3):030927, 2014. 5

[35] I. Gaiser, S. Schulz, A. Kargov, H. Klosek, A. Bierbaum, C. Pylatiuk, R. Oberle, T. Werner, T. Asfour, G. Bretthauer, and R. Dillmann. A new anthropomorphic robotic hand. *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 418–422, 2008. 16

[36] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi. Control strategies for soft robotic manipulators: A survey. *Soft Robotics*, 5(2):149–163, 2018. doi: 10.1089/soro.2017.0007. URL https://doi.org/10.1089/soro.2017.0007. PMID: 29297756. 6

[37] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995. 21

[38] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi. Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. *IEEE Transactions on Robotics*, 31(4):823–834, Aug 2015. ISSN 1552-3098. doi: 10.1109/TRO.2015.2428511. 6

[39] F. Glover and K. Sörensen. Metaheuristics. *Scholarpedia*, 10(4):6532, 2015. doi: 10.4249/scholarpedia.6532. 20

[40] B. Gorissen, D. Reynaerts, S. Konishi, K. Yoshida, J.-W. Kim, and M. De Volder. Elastic inflatable actuators for soft robotic applications. *Advanced Materials*, 29(43):1604977, 2017. 4

[41] A. Gustus, G. Stillfried, J. Visser, H. Jörntell, and P. van der Smagt. Human hand modelling: kinematics, dynamics, applications. *Biological Cybernetics*, 106(11):741–755, Dec 2012. ISSN 1432-0770. doi: 10.1007/s00422-012-0532-4. URL https://doi.org/10.1007/s00422-012-0532-4. 10

[42] B. Hartke. Global optimization. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(6):879–887, 2011. doi: 10.1002/wcms.70. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.70. 20

[43] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97. URL http://dx.doi.org/10.1093/biomet/57.1.97. 21

[44] J. Hiller and H. Lipson. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466, April 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2172702. 7

[45] G. S. Hornby and J. B. Pollack. The advantages of generative grammatical encodings for physical design. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 1, pages 600–607 vol. 1, May 2001. doi: 10.1109/CEC.2001.934446. 7

[46] R. Horst and P. M. Pardalos. *Handbook of global optimization*, volume 2. Springer Science & Business Media, 2013. 17

[47] K. Hoshino and I. Kawabuchi. Pinching with finger tips in humanoid robot hand. In *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*, pages 705–712. IEEE, 2005. 15

[48] Z. Huang, L. Zhang, S. Cheng, J. Zhang, and X. Xia. Back-analysis and parameter identification for deep excavation based on pareto multiobjective optimization. *Journal of Aerospace Engineering*, 28(6):A4014007, 2014. 18

[49] T. Iberall. Grasp planning from human prehension. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence. Milan, Italy, August 23-28, 1987*, pages 1153–1156, 1987. URL `http://ijcai.org/Proceedings/87-2/Papers/114.pdf`. 11

[50] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides. Soft robotics for chemists. *Angewandte Chemie International Edition*, 50(8):1890–1895, 2011. doi: 10.1002/anie.201006464. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201006464`. 5

[51] J. M. Inouye and F. J. Valero-Cuevas. Anthropomorphic tendon-driven robotic hands can exceed human grasping capabilities following optimization. *The International Journal of Robotics Research*, 33(5):694–705, 2014. doi: 10.1177/0278364913504247. URL `https://doi.org/10.1177/0278364913504247`. 7

[52] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers. Design of the utah/mit dextrous hand. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1520–1532. IEEE, 1986. 15

[53] D. Jeong and K. Lee. Design and analysis of an origami-based three-finger manipulator. *Robotica*, 36(2):261–274, 2018. doi: 10.1017/S0263574717000340. 5

[54] H. Jiang, Z. Wang, X. Liu, X. Chen, Y. Jin, X. You, and X. Chen. A two-level approach for solving the inverse kinematics of an extensible soft arm considering viscoelastic behavior. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE International Conference, 2017. 6

[55] M. Kaltenbrunner, T. Sekitani, J. Reeder, T. Yokota, K. Kuribara, T. Tokuhara, M. Drack, R. Schwödiauer, I. Graz, S. Bauer-Gogonea, et al. An ultra-lightweight design for imperceptible plastic electronics. *Nature*, 499(7459):458, 2013. 4

[56] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi, and Y. Miura. Patterns of static prehension in normal hands. *American Journal of Occupational Therapy*, 34(7):437–445, 1980. 12

[57] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi, and Y. Miura. Patterns of static prehension in normal hands. *American Journal of Occupational Therapy*, 34(7):437–445, 1980. doi: 10.5014/ajot.34.7.437. URL `+http://dx.doi.org/10.5014/ajot.34.7.437`. 10, 11

[58] S. B. Kang and K. Ikeuchi. A framework for recognizing grasps. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1991. 12

[59] I. A. Kapandji. *Funktionelle Anatomie der Gelenke: schematisierte und kommentierte Zeichnungen zur menschlichen Biomechanik*. Georg Thieme Verlag, 2009. 12

[60] R. K. Katzschmann, A. D. Marchese, and D. Rus. Autonomous object manipulation using a soft planar grasping manipulator. *Soft Robotics*, 2(4):155–164, 2015. doi: 10.1089/soro.2015.0013. URL https://doi.org/10.1089/soro.2015.0013. PMID: 27625916. 5

[61] H. Kawasaki, T. Komatsu, and K. Uchiyama. Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii. *IEEE/ASME transactions on mechatronics*, 7(3):296–303, 2002. 15

[62] S. Kim, C. Laschi, and B. Trimmer. Soft robotics: a bioinspired evolution in robotics. *Trends in Biotechnology*, 31(5):287 – 294, 2013. ISSN 0167-7799. doi: https://doi.org/10.1016/j.tibtech.2013.03.002. URL http://www.sciencedirect.com/science/article/pii/S0167779913000632. 7

[63] D. Kozen. *Theory of Computation*. Texts in Computer Science. Springer London, 2006. ISBN 9781846282973. 18

[64] J. L. Lagrange. *Théorie des fonctions analytiques*. Impr. de la République, 1797. 17

[65] E. Landau. Handbuch der lehre von der verteilung der primzahlen. *Bull. Amer. Math. Soc*, 1909. 17

[66] J. M. Landsmeer. Power grip and precision handling. *Annals of the rheumatic diseases*, 21: 164–70, 1962. 11

[67] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez. Real-time control of soft-robots using asynchronous finite element modeling. *ICRA*, 2015. URL https://hal.inria.fr/hal-01163760/document. 14

[68] C. Larson, B. Peele, S. Li, S. Robinson, M. Totaro, L. Beccai, B. Mazzolai, and R. Shepherd. Highly stretchable electroluminescent skin for optical signaling and tactile sensing. *Science*, 351 (6277):1071–1074, 2016. 4

[69] L. Liberti. Introduction to global optimization. *Ecole Polytechnique*, 2008. URL https://www.lix.polytechnique.fr/~liberti/teaching/globalopt-lima.pdf. Accessed: August 11, 2018. 20

[70] A. C. Lin and N. H. Quang. Automatic generation of mold-piece regions and parting curves for complex cad models in multi-piece mold design. *Computer-Aided Design*, 57:15 – 28, 2014. ISSN 0010-4485. doi: https://doi.org/10.1016/j.cad.2014.06.014. URL http://www.sciencedirect.com/science/article/pii/S0010448514001420. 27, 50

[71] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. Jin, Y. Liu, S. Fan, T. Lan, and Z. Chen. Multisensory five-finger dexterous hand: The dlr/hit hand ii. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3692–3697. IEEE, 2008. 15

[72] C. Lovchik and M. A. Diftler. The robonaut hand: A dexterous robot hand for space. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 907–912. IEEE, 1999. 15

[73] R. Ma and A. Dollar. Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption. *IEEE Robotics Automation Magazine*, 24(1):32–40, March 2017. ISSN 1070-9932. doi: 10.1109/MRA.2016.2639034. 5

[74] A. D. Marchese and D. Rus. Design, kinematics, and control of a soft spatial fluidic elastomer manipulator. *The International Journal of Robotics Research*, 35(7):840–869, 2016. doi: 10.1177/0278364915587925. URL https://doi.org/10.1177/0278364915587925. 6

[75] A. D. Marchese, K. Komorowski, C. D. Onal, and D. Rus. Design and control of a soft and continuously deformable 2d robotic manipulation system. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2189–2196, May 2014. doi: 10.1109/ICRA.2014.6907161. 5, 6

[76] R. V. Martinez, J. L. Branch, C. R. Fish, L. Jin, R. F. Shepherd, R. M. D. Nunes, Z. Suo, and G. M. Whitesides. Robotic tentacles with three-dimensional mobility based on flexible elastomers. *Advanced Materials*, 25(2):205–212, 2012. doi: 10.1002/adma.201203002. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201203002`. 5

[77] J. McCann, L. Albaugh, V. Narayanan, A. Grow, W. Matusik, J. Mankoff, and J. Hodgins. A compiler for 3d machine knitting. *ACM Transactions on Graphics (TOG)*, 35(4):49, 2016. 28, 50

[78] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949. 21

[79] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. 21

[80] S. Mghames, M. Laghi, C. D. Santina, M. Garabini, M. Catalano, G. Grioli, and A. Bicchi. Design, control and validation of the variable stiffness exoskeleton flexo. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 539–546, July 2017. doi: 10.1109/ICORR.2017.8009304. 4

[81] B. Mosadegh, P. Polygerinos, C. Keplinger, S. Wennstedt, R. F. Shepherd, U. Gupta, J. Shim, K. Bertoldi, C. J. Walsh, and G. M. Whitesides. Pneumatic networks for soft robotics that actuate rapidly. *Advanced Functional Materials*, 24(15):2163–2170, 2014. URL `http://dx.doi.org/10.1002/adfm.201303288`. 4, 5

[82] R. Murray. *A Mathematical Introduction to Robotic Manipulation*. Taylor & Francis Limited, 2017. ISBN 9781138440166. 14, 16

[83] R. Mutlu, G. Alici, M. in het Panhuis, and G. M. Spinks. 3d printed flexure hinges for soft monolithic prosthetic fingers. *Soft Robotics*, 3(3):120–133, 2016. 5

[84] Y. C. Nakamura, D. M. Troniak, A. Rodriguez, M. T. Mason, and N. S. Pollard. The complexities of grasping in the wild. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 233–240, Nov 2017. doi: 10.1109/HUMANOIDS.2017.8246880. 14

[85] J. R. Napier. The prehensile movements of the human hand. *The Journal of bone and joint surgery. British volume*, 38(4):902–913, 1956. 10, 11

[86] A. Neumann. Constraint satisfaction and global optimization in robotics, 2003. URL `http://www.mat.univie.ac.at/~neum/ms/robslides.pdf`. Accessed: August 10th 2018. 18, 19

[87] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar. A compliant, underactuated hand for robust manipulation. *The International Journal of Robotics Research*, 33(5):736–752, 2014. doi: 10.1177/0278364913514466. URL `https://doi.org/10.1177/0278364913514466`. 5

[88] P. Ohta, L. Valle, J. King, K. Low, J. Yi, C. G. Atkeson, and Y.-L. Park. Design of a lightweight soft robotic arm using pneumatic artificial muscles and inflatable sleeves. *Soft Robotics*, 5(2):204–215, 2018. doi: 10.1089/soro.2017.0044. URL `https://doi.org/10.1089/soro.2017.0044`. PMID: 29648951. 4

[89] T. Okada. Object-handling system for manual industry. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(2):79–89, 1979. 15

[90] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Mass. [u.a.], 1994. ISBN 0-201-53082-1. URL http://www.gbv.de/dms/ilmenau/toc/12708035X.PDF. 18

[91] Y. Park, B. Chen, and R. J. Wood. Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors. *IEEE Sensors Journal*, 12(8):2711–2718, Aug 2012. ISSN 1530-437X. doi: 10.1109/JSEN.2012.2200790. 4

[92] C. J. Payne, I. Wamala, C. Abah, T. Thalhofer, M. Saeed, D. Bautista-Salinas, M. A. Horvath, N. V. Vasilyev, E. T. Roche, F. A. Pigula, and C. J. Walsh. An implantable extracardiac soft robotic device for the failing heart: Mechanical coupling and synchronization. *Soft Robotics*, 4 (3):241–250, 2017. URL https://doi.org/10.1089/soro.2016.0076. 4

[93] R. Pelrine, R. Kornbluh, Q. Pei, and J. Joseph. High-speed electrically actuated elastomers with strain greater than 100%. *Science*, 287(5454):836–839, 2000. ISSN 0036-8075. doi: 10.1126/science.287.5454.836. URL http://science.sciencemag.org/content/287/5454/836. 4

[94] N. S. Pollard and V. B. Zordan. Physically based grasping control from example. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 311–318, New York, NY, USA, 2005. ACM. ISBN 1-59593-198-8. doi: 10.1145/1073368.1073413. URL http://doi.acm.org/10.1145/1073368.1073413. 10

[95] P. Polygerinos, S. Lyne, Z. Wang, L. F. Nicolini, B. Mosadegh, G. M. Whitesides, and C. J. Walsh. Towards a soft pneumatic glove for hand rehabilitation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 3-7 Nov. 2013. URL http://dx.doi.org/10.1109/IROS.2013.6696549. 4

[96] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh. Soft robotic glove for combined assistance and at-home rehabilitation. *Robotics and Autonomous Systems*, 73:135 – 143, 2015. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2014.08.014. URL http://www.sciencedirect.com/science/article/pii/S0921889014001729. Wearable Robotics. 4

[97] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. ISBN 0521880688, 9780521880688. 22

[98] J. Reddy. *Introduction to the Finite Element Method*. McGraw-Hill, 1993. ISBN 9780070513563. URL https://books.google.de/books?id=2NOlPwAACAAJ. 15

[99] J. Rieffel, D. Knox, S. Smith, and B. Trimmer. Growing and evolving soft robots. *Artificial Life*, 20(1):143–162, 2014. doi: 10.1162/ARTL_a_00101. URL https://doi.org/10.1162/ARTL_a_00101. PMID: 23373976. 6, 7

[100] M. J. P. F. Ritt. Lister's the hand. diagnosis and indications. *Journal of Hand Surgery*, 27(4): 397–397, 2002. doi: 10.1054/jhsb.2002.0757. URL https://doi.org/10.1054/jhsb.2002.0757. 12, 30

[101] M. Rolf and J. J. Steil. Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1147–1160, June 2014. ISSN 2162-237X. doi: 10.1109/TNNLS.2013.2287890. 6

[102] J. Romero. 12

[103] D. Rus and M. T. Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467, 2015. 4, 42

[104] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003. doi: 10.1137/1.9780898718003. URL `https://epubs.siam.org/doi/abs/10.1137/1.9780898718003`. 17

[105] J. K. Salisbury and B. Roth. Kinematic and force analysis of articulated mechanical hands. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105(1):35–41, 1983. 15

[106] F. Saunders, B. A. Trimmer, and J. Rife. Modeling locomotion of a soft-bodied arthropod using inverse dynamics. *Bioinspiration & Biomimetics*, 6(1):016001, 2011. URL `http://stacks.iop.org/1748-3190/6/i=1/a=016001`. 6

[107] J. Sauppe, editor. *Mathematical Programming Glossary*. INFORMS Computing Society, `http://glossary.computing.society.informs.org`, 2006–18. Originally authored by Harvey J. Greenberg, 1999-2006. 17

[108] C. Schlagenhauf. Interactive design and control of tendon-driven soft foam robot hands. Master's thesis, Karlsruhe Institute of Technology, 2018. 5, 7, 8, 23, 24, 25, 27, 28, 42, 43, 49, 50

[109] C. Schlagenhauf, D. Bauer, K.-H. Chang, D. Moro, S. Coros, and N. Pollard. Control of tendon-driven soft foam robot hands. In *2018 IEEE RAS International Conference on Humanoid Robots*, 2018. Accepted. 7

[110] S. Schulz, C. Pylatiuk, M. Reischl, J. Martin, R. Mikut, and G. Bretthauer. A hydraulically driven multifunctional prosthetic hand. *Robotica*, 23(3):293–299, 2005. 16

[111] M. Shahinpoor and K. J. Kim. Ionic polymer-metal composites: I. fundamentals. *Smart materials and structures*, 10(4):819, 2001. 4

[112] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides. Multigait soft robot. *Proceedings of the National Academy of Sciences*, 108(51):20400–20403, 2011. 5

[113] J. Shintake, S. Rosset, B. Schubert, D. Floreano, and H. Shea. Versatile soft grippers with intrinsic electroadhesion based on multifunctional polymer actuators. *Advanced Materials*, 28(2):231–238, 2015. doi: 10.1002/adma.201504264. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201504264`. 4

[114] J. Shintake, V. Cacucciolo, D. Floreano, and H. Shea. Soft robotic grippers. *Advanced Materials*, 30(29):1707035, 2018. doi: 10.1002/adma.201707035. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201707035`. 4

[115] H. Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41 (2):11:1–11:36, Feb. 2015. ISSN 0098-3500. doi: 10.1145/2629697. URL `http://doi.acm.org/10.1145/2629697`. 31

[116] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 29

[117] S. K. Skerik, M. Weiss, and A. E. Flatt. Functional evaluation of congenital hand anomalies. *The American journal of occupational therapy : official publication of the American Occupational Therapy Association*, 25 2:98–104, 1971. 11

[118] J. A. Snyman and D. N. Wilke. *Practical Mathematical Optimization*. Springer International Publishing, 2 edition, 2018. ISBN 978-3-319-77586-9. doi: 10.1007/978-3-319-77586-9. 17

[119] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pages 175–184, New York, NY, USA, 2004. ACM. ISBN 3-905673-13-4. doi: 10.1145/1057432.1057456. URL http://doi.acm.org/10.1145/1057432.1057456. 31

[120] T. G. Thuruthel, E. Falotico, M. Cianchetti, and C. Laschi. Learning global inverse kinematics solutions for a continuum robot. In V. Parenti-Castelli and W. Schiehlen, editors, *ROMANSY 21 - Robot Design, Dynamics and Control*, pages 47–54, Cham, 2016. Springer International Publishing. ISBN 978-3-319-33714-2. 6

[121] L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4): 1701–1728, 1994. ISSN 00905364. URL http://www.jstor.org/stable/2242477. 21

[122] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr. Xsede: Accelerating scientific discovery. *Computing in Science & Engineering*, 16(5):62–74, Sept.-Oct. 2014. ISSN 1521-9615. doi: 10.1109/MCSE.2014.80. URL doi.ieeecomputersociety.org/10.1109/MCSE.2014.80. 36

[123] B. Walsh. Markov chain monte carlo and gibbs sampling. *Lecture notes for EEB 596, University of Arizona*, 596, 2002. URL http://nitro.biosci.arizona.edu/courses/EEB596/handouts/Gibbs.pdf. Accessed: August 14th 2018. 22

[124] T. Weise, M. Zapf, R. Chiong, and A. J. Nebro. *Nature-inspired algorithms for optimisation*, volume 193. Springer, 2009. doi: 10.1007/978-3-642-00267-0. 18

[125] J. Wirekoh and Y.-L. Park. Design of flat pneumatic artificial muscles. *Smart Materials and Structures*, 26(3):035009, 2017. URL http://stacks.iop.org/0964-1726/26/i=3/a=035009. 4

[126] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997. 18

[127] D. H. Wolpert, W. G. Macready, et al. No free lunch theorems for search. Technical report, Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995. 18

[128] Z. Xu and E. Todorov. Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3485–3492, 2016. 5, 6

[129] H. K. Yap, H. Y. Ng, and C.-H. Yeow. High-force soft printable pneumatics for soft robotic applications. *Soft Robotics*, 3(3):144–158, 2016. 4

[130] Z. Zhou, S. Gao, Z. Gu, and J. Shi. A feature-based approach to automatic injection mold generation. In *Proceedings Geometric Modeling and Processing 2000. Theory and Applications*, pages 57–68, April 2000. doi: 10.1109/GMAP.2000.838238. 27