Karlsruhe Institute of Technology

Carnegie Mellon University

Interactive Design and Control of Tendon-Driven Soft Foam Robot Hands

Master's Thesis
of

Cornelia Schlagenhauf

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

Cornelia Schlagenhauf

Abstract

Many robotic hands have been developed over the last decades - yet we are still far away from robots that are able to grasp objects, dexterously manipulate them and safely interact with humans. Although robotics research and particularly the fields of computer vision and artificial intelligence have advanced rapidly, state-of-the-art robotic hands still fall short compared to their human counterpart. The emerging field of soft robotics employs compliant materials to build and control robots which are inherently safe and promise to hold great potential to create and control dexterous hands. However, compliant materials are difficult to model, and new tools and methods must be developed to aid the design and control of soft robotic hands.

This work presents the overall system and a series of tools and methods to design, fabricate and operate a new class of robotic hands: tendon-driven soft foam hands.

Apart from being truly soft, a major goal for this type of robot is accessibility. If we want robotic hands to overcome their current confinement to research labs and cages in factories, they have to be safe, affordable, easy to build and straightforward to control. In the scope of this work, easy fabrication techniques are presented, and an interactive simulation framework for predicting hand deformations for different hand morphologies and tendon patterns is developed and evaluated. A modular control framework is created, and grasping and manipulation abilities of foam robot hands are demonstrated. Additionally, data-driven approaches to learn the inverse kinematics of foam hands are presented, which can be learned using either the physical or the simulated hand.

Kurzzusammenfassung

In den letzten Jahrzehnten wurde eine Vielzahl von Roboterhänden entwickelt - und doch sind wir noch weit entfernt von Robotern, die Objekte sicher greifen und manipulieren können und Sicherheit im Umgang mit Menschen garantieren. Obwohl die Robotikforschung große Fortschritte gemacht hat, insbesondere in den Bereichen maschinelles Sehen und künstliche Intelligenz, erreichen Roboterhände vom aktuellen Stand der Technik noch lange nicht die Fähigkeiten der menschlichen Hand. Das Gebiet der Weichrobotik, „Soft Robotics", stellt ein gerade entstehendes, aufstrebendes Feld in der Robotik dar. Weichroboter bestehen aus nachgiebigen Materialien, sind daher sicher im Umgang mit Personen und haben großes Potenzial für die Entwicklung vielseitiger Roboterhände.

Im Rahmen dieser Arbeit wurde ein Gesamtsystem entwickelt für das Design, die Fertigung und die Anwendung eines neuen Roboterhandtyps: sehnengetriebene, weiche Schaumstoffroboterhände. Die entwickelten Hände bestehen ausschließlich aus weichen Materialien. Ein weiteres Ziel dieser Entwicklung ist das Design, die Fertigung, und Anwendung von Roboterhänden möglichst zugänglich zu gestalten, um deren momentane Beschränkung auf Forschungslabore und Käfige in Indutrieanlagen zu überwinden. Dafür müssen Roboterhände sicher und einfach herzustellen und simpel zu steuern sein.

Diese Arbeit stellt entsprechend einfache Fertigungsmethoden vor, und präsentiert und evaluiert eine interaktive Simulationsumgebung, mit der Verformungen der weichen Hand für verschiedene Handmorphologien und Sehnen simuliert werden können. Eine modulare Steuerungssoftware wird entwickelt und die Fähigkeiten der Schaumstoffroboterhände in Bezug auf Greifen und Manipulation werden aufgezeigt. Desweiteren werden verschiedene datengetriebene Ansätze vorgestellt um die inverse Kinematik der Hände zu lernen, einerseits am physischen Roboter und andererseits auf Basis der Simulation.

Acknowledgments

Firstly, I would like to thank Nancy Pollard at Carnegie Mellon University, who agreed to supervise this thesis work and continuously supported my study and research, for her guidance, motivation and support. I would also like to thank Tamim Asfour at the H^{2}T, who encouraged me to apply for the exchange program with CMU, supported and also advised me during this work and always provided valuable feedback and inspiration. I also thank Ralf Mikut at the Institute for Automation and Applied Informatics (IAI). He always had an open door for me if I had a question, and provided insightful comments and ideas. Ralf Mikut was also especially flexible and forthcoming about officially supervising this thesis as part of the mechanical engineering department.

I would like to express my gratitude to Julia Starke for providing valuable feedback, comments and ideas.

I am also grateful for the helpful advice and guidance by Stelian Coros and James Bern. Special thanks go to my co-workers in the lab: Jonathan King, Kai-Hung Chang and Yuzuko Nakamura, who helped creating foam hands.

I would also like to thank my family for always supporting and encouraging me to pursue my dreams and goals.

I thank Dominik Bauer for always supporting and motivating me, spending many hours discussing problems in detail and working with me on solving them, and for always being there for me.

The research conducted towards this thesis was mainly executed at the Robotics Institute at Carnegie Mellon University as part of the Continuous Learning in International Collaborative Studies (CLICS) exchange program. I am thankful for this opportunity and thank the organizers of this program, participating in this program enabled me to grow professionally and personally.

Contents

1. Introduction 3

2. Fundamentals 4
2.1. Robot Kinematics 4
2.1.1. Position and Orientation in 3D Space 4
2.1.2. Forward and Inverse Kinematics 5
2.2. Grasping and Manipulation 5
2.2.1. Human Hand Anatomy 5
2.2.2. Grasp Properties and Contact Models 6
2.2.3. Grasp Taxonomies 8
2.2.4. Manipulation Taxonomies 11
2.2.5. Control of Grasping 11
2.3. Software Frameworks 12
2.3.1. Network Communication 12
2.3.2. Robot Operating System (ROS) 13
2.4. Sensors 13
2.4.1. Vicon Motion Capture 13
2.4.2. CyberGlove 14
2.5. Finite Element Simulation 14
2.6. Machine Learning 15
2.6.1. Supervised Learning 15
2.6.2. Reinforcement Learning 17

3. Related Works 18
3.1. Robotic Hands 18
3.2. Design and Fabrication of Soft Robots 19
3.2.1. Materials and Actuation 19
3.2.2. Design and Fabrication 21
3.3. Modeling and Control Strategies for Soft Robots 22
3.3.1. Modeling 22
3.3.2. Control 22
3.4. Challenges and Contributions 23

4. Soft Foam Robot Hands 25

5. Fabrication of Soft Foam Robot Hands 26
5.1. Manufacturing Foam Robot Hands 26
5.2. Extension and Improvements 28
5.3. Effects of Different Tendon Routings and Rest Pose 30

6. Interactive Design and Simulation Framework 32
6.1. Modeling Hand Deformation 32
6.2. Simulation Modules 34
6.2.1. Interactive Creation of Tendon Routings 34
6.2.2. Interaction with Physical Robot Hand 35
6.2.3. Fingertip Control and Pose Generation using a Cyberglove 35

6.3. Validation using Vicon Motion Capture . . . 36
6.4. Application: Human Subject Study on Tendon Routing Design Ability . . . 39
6.4.1. Questions . . . 39
6.4.2. Experiments . . . 40
6.4.3. Results and Discussion . . . 41
6.4.4. Limitations and Outlook . . . 42

7. Operation of Tendon-Driven Soft Foam Robot Hands 44
7.1. Modular ROS Control Framework . . . 44
7.2. Experiments: Repeatability, Strength, Grasping and In-hand Manipulation . . . 46
7.3. Weaknesses and Current Limitations . . . 49
7.4. Teleoperation . . . 49
7.5. Learning in Simulation: Inverse Kinematics . . . 51
7.5.1. Learning-based methods . . . 51
7.5.2. Experiments . . . 51
7.5.3. Results and Discussion . . . 52

8. Conclusion 53
8.1. Summary of Results . . . 53
8.2. Future Work . . . 54
8.2.1. Materials and Fabrication . . . 54
8.2.2. Sensors . . . 54
8.2.3. Closed-loop Control . . . 54
8.2.4. Contact Simulation . . . 54
8.2.5. Learning Manipulations . . . 55

A. Appendix 56
A.1. Principal Component Analysis . . . 56
A.2. A* Path Planning . . . 56
A.3. User Study . . . 59
A.3.1. Lateral Tripod . . . 59
A.3.2. Medium Wrap a . . . 60
A.3.3. Prismatic 3-finger . . . 61
A.3.4. Average error over all grasps . . . 62

1. Introduction

Conventional robots are built from rigid materials to achieve precise and predictable behavior. This enables the usage of relatively easy modeling and control approaches, but as a safe interaction with humans cannot be guaranteed, it also limits their application to strictly separated spaces such as cages in factories, or research labs.

To overcome this limitation and bridge the gap between machines and humans, the research field of soft robotics uses compliant materials and develops new techniques to build and control safe robots. Apart from safe interaction with humans and handling of delicate objects, a further motivation for soft structures in robotics is the observation that biological systems which demonstrate softness and body compliance often exceed rigid machines in their capabilities. A variety of plants and animals are able to perform complex movements with soft structures and adapt easily to object shapes in grasping, e.g. the elephant trunk or octopus arms.

Robotic grasping and manipulation technologies have advanced rapidly over the last decades, and could be used to solve a variety of pressing issues in our world. Two examples of recent problems where robots could potentially be helpful are the shortages of workers in the agricultural and in the care sector.

These tasks could be carried out by robots - if they had the appropriate hands. As an example, picking soft fruits and vegetables may seem to be a simple task. However, in addition to the unstructured and constantly changing environment, the large variety in shape, size and order of ripeness of fruits are a big challenge for conventional robotic grippers, and require careful design of gripper hardware and control schemes.

Particularly if we want robots to interact with people or handle delicate objects, and robustly perform grasps and manipulations, the concept of soft hands offers an interesting approach towards the development of easy-to-control, compliant and dexterous manipulators. In order to be accessible, these hands must meet a set of criteria:

They must be inherently **soft**, the hands must be compliant enough to cause no harm to people, delicate objects or their environment. To be affordable for everyone who wishes to experiment with them and use them, soft hands have to be **inexpensive**. Depending on the underlying task we want to use a hand for, different requirements can be identified for a hand design (e.g. a simple pick and place task requires a basic gripper, while a complex in-hand manipulation requires a more sophisticated hand design). Therefore, accessible soft robot hands must be **customizable** and be tailored to the desired task.

This work presents methods for the design, fabrication and operation of tendon-driven soft foam robot hands, a new class of soft robots which promise to be low-cost, customizable and accessible to non-experts. Since the main body of these robots consists of flexible foam they are termed ’foam robots’. A foam robot is moved by tendons which are sewn through a textile skin on the robot, and driven by servo mounted winches.

Chapter 2 describes mathematical principles, methods and hardware used in this work, and introduces the fundamentals of grasping and manipulation. Insights on the nature of human hands and how we use them are fundamental to understanding the challenges a robotic hand system must face.

Related works concerned with the design, fabrication and control of soft robotic hands are presented in Chapter 3.

Chapter 4 introduces the concept of tendon-driven soft foam robot hands and Chapter 5 details their fabrication. A simulation framework is developed to predict deformation and behavior of these hands and described in Chapter 6. Methods to operate foam hands are explored in Chapter 7, and Chapter 8 summarizes the results of this work and presents approaches to be addressed in future works.

2. Fundamentals

In order to facilitate understanding the nature and challenges of soft robotic grasping and manipulation, this chapter reviews fundamental principles, mechanisms and techniques that are used to describe, build and control soft hands.

2.1. Robot Kinematics

This section summarizes essential principles and techniques to describe movements of robotic systems. A brief overview of methods to express position and orientation in 3D space will be followed by a description of the forward and inverse kinematics problem in rigid manipulators, which has been extensively studied by researchers. In contrast to rigid manipulators, soft robots possess great compliance and/or a deformable nature, which significantly complicates the study of their kinematics.

2.1.1. Position and Orientation in 3D Space

Positions in 3D space are denoted by a 3x1 vector ${}^j\mathbf{p} = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}^T$ relative to an inertial coordinate frame j. All coordinate frames in this section are right-handed. To express rotations, several representations exist, however this section will only introduce two of them, rotation matrices and quaternions. For a more detailed description and a summary of more representations of rotation (e.g. Euler angles, axis-angle) see [77, 107].

Rotation matrices: The three base vectors of a coordinate frame i are expressed in the inertial coordinate frame j: ${}^j\mathbf{x}_i, {}^j\mathbf{y}_i, {}^j\mathbf{z}_i \in \mathbb{R}^3$. Writing them together yields a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$,

$$\mathbf{R} = \begin{pmatrix} {}^j\mathbf{x}_i & {}^j\mathbf{y}_i & {}^j\mathbf{z}_i \end{pmatrix} \tag{2.1}$$

with the columns of $\mathbf{R}$ being mutually orthonormal, and the determinant 1, $\det \mathbf{R} = 1$. The rotation matrix ${}^j\mathbf{R}_i$ transforms a vector expressed in the coordinate frame i to a vector in the coordinate frame j. It can therefore be used to express rotations between frames. A sequence of rotations can be obtained by simple matrix multiplication: ${}^j\mathbf{R}_k = {}^j\mathbf{R}_i{}^i\mathbf{R}_k$.

Quaternions: A quaternion formally is a vector quantity of the form

$$\mathbf{q} = a + u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k} \tag{2.2}$$

where a is referred to as the scalar component, and $\mathbf{u} = (u_1, u_2, u_3)$ represents the vector component. Quaternions are often expressed in the shorthand notation $\mathbf{q} = (a, \mathbf{u})$. The *conjugate* of a quaternion $\mathbf{q}$ is given by $\mathbf{q}^* = (a, -\mathbf{u})$, and $\mathbf{q}$ is a *unit quaternion*, if $||\mathbf{q}||^2 = \mathbf{q} \cdot \mathbf{q}^* = 1$.

For a unit quaternion $\mathbf{q} = (a, \mathbf{u})$ and a vector $\mathbf{p}$, the operation $\mathbf{qpq}^*$ rotates $\mathbf{p}$ about the axis $\mathbf{u}$. Given a desired axis $\mathbf{w}$ and angle θ for a rotation, the associated unit quaternion is $\mathbf{q} = (\cos\frac{\theta}{2}, \mathbf{w}\sin\frac{\theta}{2})$.

The representation of orientations with quaternions is efficient and extremely useful in robotics to avoid singularities resulting from vector/matrix representation. As an example they can be used for continuous interpolation between two orientations.

Homogeneous Transformation: The combination of position and orientation is collectively termed the *pose*. A compact notation for a given pose of a coordinate frame i relative to the coordinate frame

j, is the 4×4 homogeneous transformation matrix

$$^{j}\mathbf{T}_i = \begin{pmatrix} ^{j}\mathbf{R}_i & ^{j}\mathbf{p}_i \\ \mathbf{0} & 1 \end{pmatrix} \tag{2.3}$$

A sequence of transformations is expressed by matrix multiplication: Given the transformations $^{j}\mathbf{T}_i$,$^{i}\mathbf{T}_k$, the transformation matrix $^{j}\mathbf{T}_k$ is obtained by $^{j}\mathbf{T}_k = {}^{j}\mathbf{T}_i{}^{i}\mathbf{T}_k$. This property is especially helpful in the robotics domain to describe forward kinematics.

2.1.2. Forward and Inverse Kinematics

Traditional robotic manipulators are systems of rigid bodies (*links*), which are connected by *joints*, forming a *kinematic chain* from the robot base to the *end effector*. The term "end effector" describes the interface between a robot manipulator and the environment. The position and orientation of each link is usually described by a coordinate frame attached to the link. Several conventions have been developed to locate coordinate frames of adjacent links relative to another, with the *Denavit-Hartenberg* convention being the most widely used. Detailed explanations of methods to describe configurations of robotic systems, as well as more information on robot kinematics in general can be found in literature [25, 107, 77].

Forward kinematics refers to the problem of determining the robot end effector pose relative to the robot base for a given set of joint angles. For a serial chain, the transformation between the end effector frame and base frame can be obtained by simple concatenation of the transformations between adjacent links. The manipulator *workspace* describes the set of all end effector poses which can be reached by a choice of joint angles.

The inverse problem, finding the corresponding joint angles for a given end effector pose, is termed *inverse kinematics*. Methods to describe and solve both forward and inverse kinematics for rigid manipulators have been studied and reviewed extensively in literature (e.g. [77, 25, 107]) and are therefore not further detailed here.

While the inverse kinematics problem has been largely solved for rigid manipulators, in the emerging field of soft robotics additional problems have to be addressed. Their ability to accomplish motions such as buckling, contraction, extension or bending, results in soft robots having virtually infinite degrees of freedom. Additionally, Thuruthel et al. [113] mention nonlinear material effects such as compliance and hysteresis, as well as the wide range of design and actuation techniques that account for the non-trivial nature of this problem. Previous works have particularly studied the problem of inverse kinematics in soft robotics and are reviewed more detailed in Section 3.3.

2.2. Grasping and Manipulation

The human hand represents a highly complex mechanism and much effort has been devoted to understand and catalogue human grasping. A major goal for roboticists is to not only understand the human hand and underlying grasping techniques but to leverage those to create robotic hands capable of dexterous grasping and manipulation. This section details essential grasp properties and the structure of the human, and summarizes grasp and manipulation taxonomies.

2.2.1. Human Hand Anatomy

Figure 2.1 illustrates the 27 bones of a human hand and the joints connecting them. They are divided into the following subgroups:
The *carpals*, eight short bones, form the wrist of the hand. **Carpometacarpal (CM) joints** connect the wrist to the *metacarpals*. The *proximal phalanges* articulate with the Metacarpals at the **metacarpophalangeal (MP) joints**. *Intermediate* and *distal phalanges* are connected by **proximal (PIP)** and **distal**

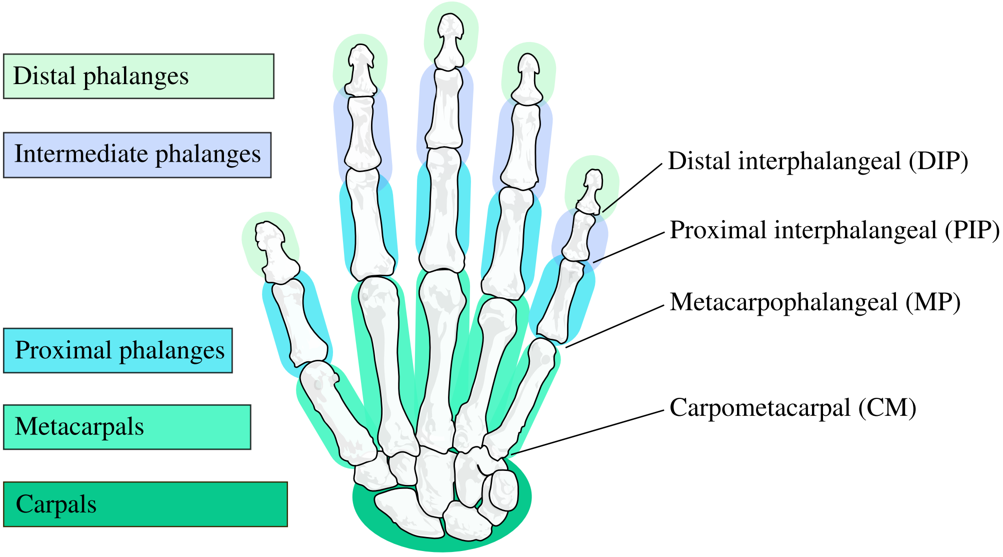


Figure 2.1.: Skeletal structure of the human hand. Bones are listed on the left, joints on the right.

interphalangeal (DIP) joints. The thumb only contains a proximal and a distal phalanx.
The motions of limbs are classified according to the anatomical plane they occur in:

Flexion/Extension: Extension describes a straightening motion wherein the angle between body parts increases. Flexion denotes the contrary movement.

Abduction/Adduction: Abduction is characterized by a movement away from the center line of the boy, whereas adduction describes motions towards the body's midline.

Rotation: Rotation refers to the movement of limbs around their long axis.

The human hand has 27 degrees of freedom (DoF): Each finger has three DoF for flexion/extension and one DoF for abduction/adduction. The thumb is more complex than the other fingers, it has 5 DoF: 3 flexion/extension and 2 abduction/adduction. This leaves 6 DoF at the wrist for rotation and translation. Depending on purpose and application, a variety of human hand models can be found in literature. They vary in kinematics and number of DoF and always represent a trade-off between performance and simplicity. Human hand models have been developed not only in the field of robotics, but also in other research areas, e.g. computer vision, biomedical engineering, human-computer interaction. Examples of human hand models are described in [86, 64, 23, 109, 112].

2.2.2. Grasp Properties and Contact Models

A grasp describes a system wherein the fingers of a robot or human hand grip a desired object. One important analytical measure to describe a grasp is *force closure*, which implies that the fingers are able to resist any arbitrary force and/or moment acting on the object externally [105]. In order to perform everyday tasks, a force closure grasp must possess certain properties which have been identified and studied by researchers [27, 68, 47]. Shimoga [105] divides them into four mutually independent properties:

- **Dexterity**: "(The) capability of changing the position and orientation of the manipulated object from a given reference configuration to a different one, arbitrarily chosen within the hand workspace", [7].
- **Equilibrium**: The resultant forces and torques applied on the grasped object (by external disturbances and fingers) is null [60, 13].

- **Stability**: A stable grasp refers to a grasped object at equilibrium point, when after an arbitrary small deflection, the grasp returns to its original position [45, 26].
- **Dynamic behavior**: "(The) time response of the grasp for changes in its motion or force trajectories", [105].

In the robotics domain, the term *grasp synthesis* generally refers to the problem of finding a suitable finger configuration for a given object, where the grasp satisfies a number of criteria for a given grasping task. For a known object (a complete geometrical object model is available) this task is usually referred to as *grasp planning*. *Grasp analysis* describes the study of grasp properties for a given set of finger properties [105].

Humans as well as robotic manipulators use contact forces to grasp and interact with objects in their environment. Modeling and controlling contacts is therefore crucial for grasping. To depict the interface between the hand and the object, three different contact models are generally used [95, 87]:

Frictionless point contact: The contact patch is very small and object and hand surfaces are slippery. Only the normal component of the translational velocity and the normal component of the contact force are transmitted. Frictional forces and moments are neglected.

Frictional point contact / hard finger: The contact patch is very small, but there is significant friction. All three translational velocity and force components are transmitted. No angular velocities or moments are transmitted.

Soft contact / soft finger: Surface friction and contact patch size are large enough to produce significant friction forces and a moment about the contact normal. All three translational velocities and the normal angular velocity component are transmitted, as well as all three translational forces and the moment about the contact normal.

Friction is usually modeled using the *Coulomb friction model*. In this model tangential force is proportional to normal force and the coefficient of proportionality is determined by the materials forming the contact. A more detailed explanation of contact models can be found in [77]

According to work by Iberall [48] and MacKenzie and Iberall [70], there are three basic directions relative to the hand coordinate frame for the hand to apply forces on the object to hold it securely, shown in Figure 2.2: *Pad Opposition* refers to a configuration where involved hand surfaces are opposed along the x-axis, a direction parallel to the palm. *Palm Opposition* occurs along the z-axis, hand surfaces are oriented perpendicular to the palm. *Side Opposition* occurs between hand surfaces along the y-axis. In order to perform a grasping task, several fingers work together as a functional unit, the Virtual Finger (VF) ([46]). Fingers or hand parts are assigned to the same virtual finger if they apply forces in a similar direction and work in unison.

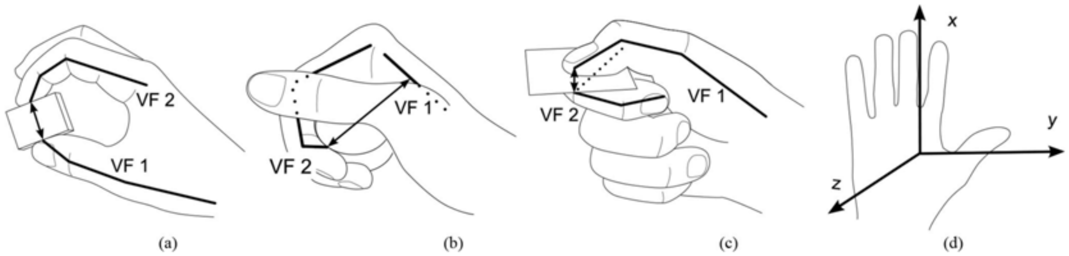


Figure 2.2.: Virtual Fingers (VF) and Opposition types: a) Pad Opposition, b) Palm Opposition, c) Side Opposition. d) shows the hand coordinate system. From [36]

2.2.3. Grasp Taxonomies

Grasp taxonomies categorize grasp types and represent benchmarks which can be used to quantify robot hand abilities (e.g. how many grasps of the taxonomy can be achieved?). Additionally, they can be useful for grasp synthesis and inspire hand designs.

Napier [80] distinguishes between two types of grasps: *precision* and *power grasps* (Figure 2.3). In the former, an object is held by the fingertips and the thumb. A power grasp in contrast is characterized by large areas of contact between the surface of the fingers and palm and the grasped object. This work also states that when grasping an object, the intended activity determines which grasp type is chosen. The distinction between power and precision grasps can be found throughout a variety of grasp taxonomies. In robotics, the most widely cited taxonomy is the Cutkosky taxonomy [27]. This work includes 16 grasp types based on observations of skilled tool use by machinists in a workshop. Within its hierarchical tree structure shown in Figure 2.4, it is distinguished between power and precision grasps in the top layer. From left to right, object size decreases and the grasps become less powerful. Grasp taxonomies regarding tasks of everyday living have been developed by Kapandji [57] and Kamakura et al. [55]. Kamakura et al. studied finger positions and contact areas human subjects demonstrated when grasping objects to perform everyday tasks. They identified 14 patterns under 4 categories (power grip, intermediate grip, precision grip and grip involving no thumb).

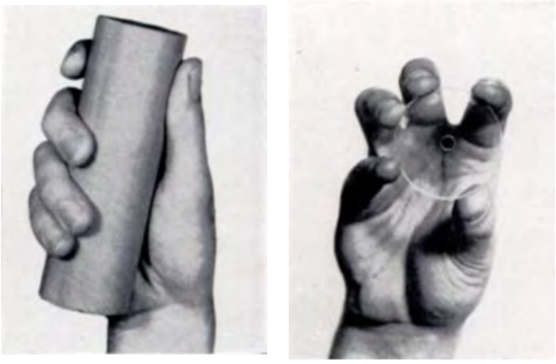


Figure 2.3.: Left: A human hand performing power grasps to hold a cylindrical object. Right: Precision grasps of a disk. From [80]

Recently, Feix et al. [36] have reviewed existing grasp taxonomies, additionally considering their own observations and synthesizing them into a single comprehensive GRASP taxonomy. They define a grasp as “(...) every static hand posture with which an object can be held securely with one hand, irrespective of the hand orientation”. This definition rules out bimanual tasks, in-hand movements and gravity dependent grasps (e.g. the hook grasp and the flat hand grasp). The taxonomy identifies a total of 33 unique prehensile grasp types.

The GRASP taxonomy was regrouped by Nancy Pollard[1] into six categories with respect to grasp type (power/precision) and considering which parts of the hand are used in the grasp:

- 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

Figure 2.5 shows the six categories and the regrouped grasps. The numbers correspond to the grasp type number in the GRASP taxonomy [36].

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

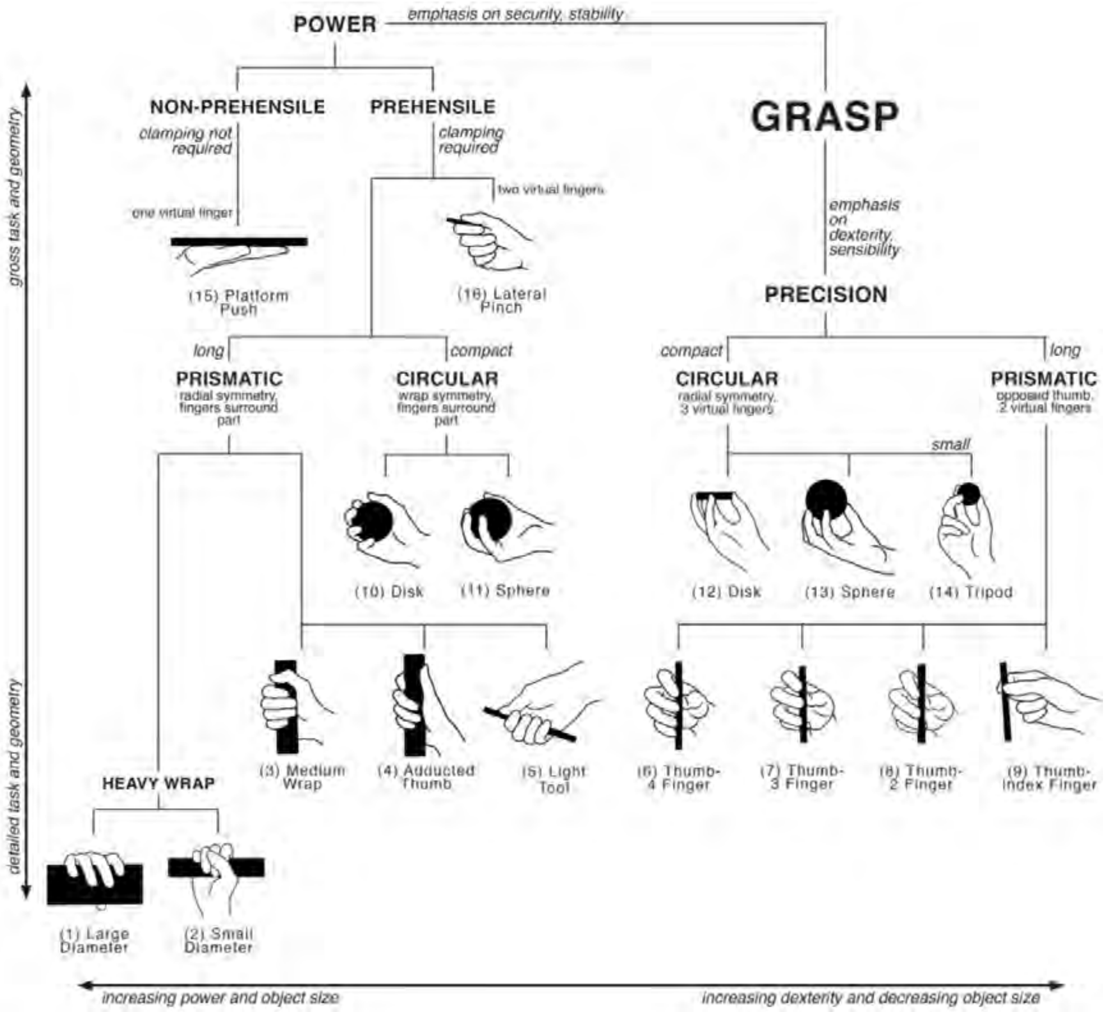


Figure 2.4.: Cutkosky taxonomy: The top level classifies power and precision grasps. Following the tree structure from left to right, object size and grasp intensity decreases. Retrieved from `https://cdn.hackaday.io/images/285241475755961545.png`.

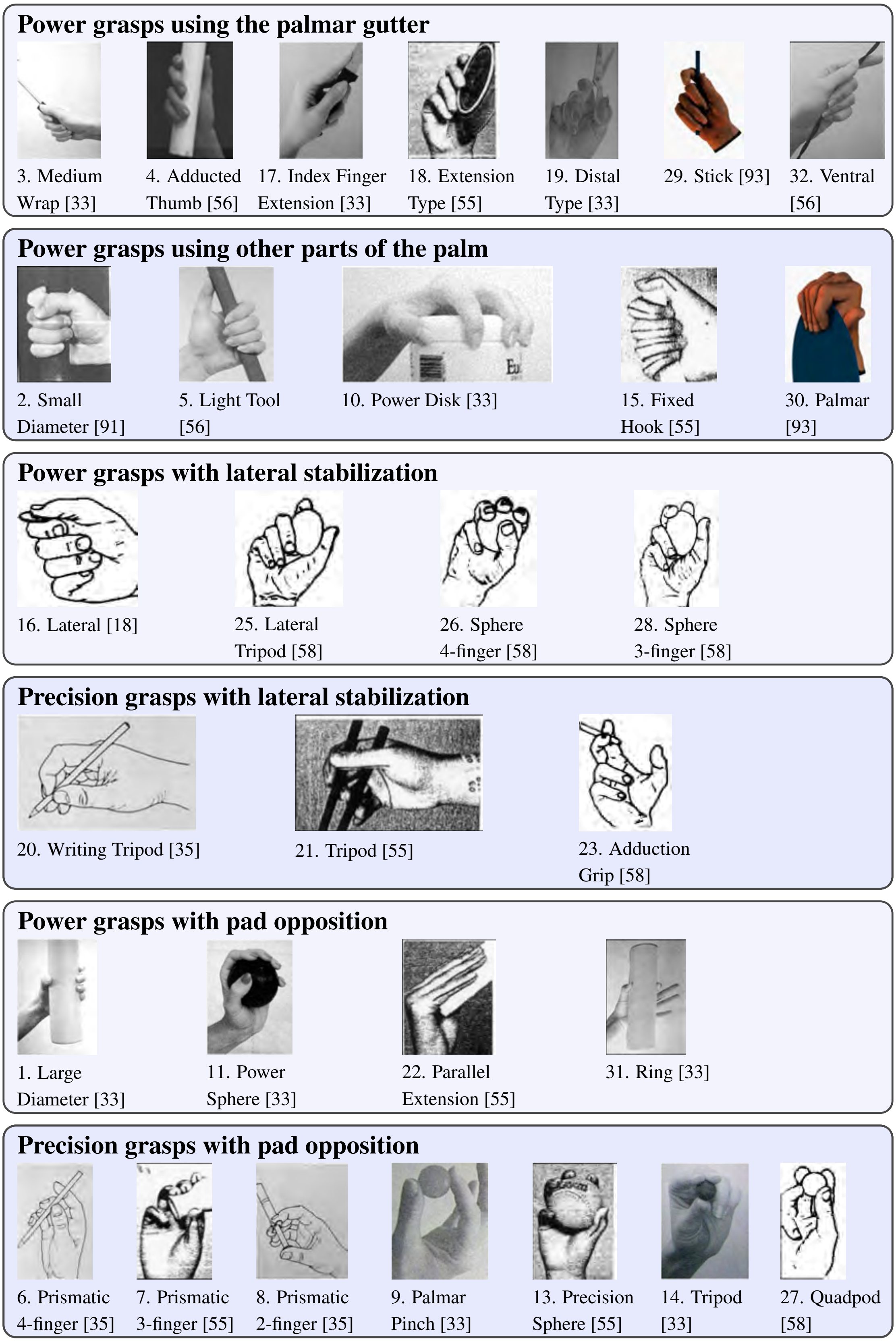


Figure 2.5.: Cumulative taxonomy by Nancy Pollard. Grasp types are divided into power and precision grasps, and grouped with respect to the parts of the hand (palmar gutter, palm, lateral finger, finger pads) that are used for the grasp. *Image by Dominik Bauer* [3].

2.2.4. Manipulation Taxonomies

Researchers have furthermore worked on characterizing and classifying manipulation actions.

Chang and Pollard [17] categorize manipulation actions before grasping. Elliott and Connolly [35] group manipulative hand movements into three categories: simple synergies (e.g. squeeze), reciprocal synergies such as roll, and sequential patterns (e.g. a stepping motion of the fingers to alter contact positions on the object in order to rotate the object).

Bullock et al. [12] provide a hand-centric and motion-centric manipulation classification, focusing on hand motion and relative motion of hand and object at contact. Their classification scheme is not limited to a specific hand design and can therefore be applied to both the human and the robotic domain.

2.2.5. Control of Grasping

Neuroscience

Despite the complex physical structure of the human hand we are able to efficiently and subconsciously choose the appropriate grasp and coordinate our hand motions. A variety of studies have been conducted to identify and understand human control strategies during grasping. Jeannerod [52, 53] analyzed the grip aperture (separation between thumb and index finger) during grasping. He found that the reach-to-grasp motion is characterized by a progressive opening of the grip followed by gradual closure of the grip until the object's size is matched. The largest grip aperture is highly correlated with object size and occurs at 60 - 70% of the reach movement, representing a distinct landmark. Castiello [15] summarizes results from various studies which have explored neurophysiology of grasping in humans and monkeys. Experiments in macaque monkeys using lesion of targeted areas in the brain show that the grasping circuit involves several areas of the monkey brain. In humans, neural grasping mechanisms and corresponding brain areas are investigated by observing grasping in patients with brain damage and noninvasive imaging studies. A grasping circuit similar to the one identified in monkeys can also be found in humans, but human grasping involves a much wider network of brain areas. In summary, these studies have advanced the knowledge on neural grasping mechanisms but fully understanding the human grasping circuit will require further experiments and analysis.

Postural Synergies and Underactuation

Neuroscientific studies suggest that during grasping the joints of the human hand are not controlled individually. Instead, "Experimental evidence indicates that the simultaneous motion and force of the fingers are characterized by coordination and covariation patterns that reduce the number of independent degrees of freedom to be controlled" (Bicchi et al. [8]).

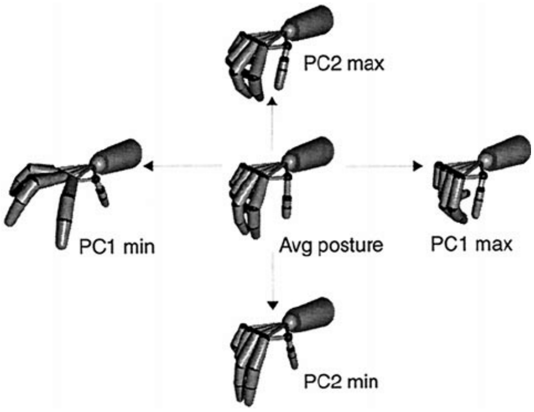


Figure 2.6.: Postural synergies given by the first two principal components (PC1, PC2). From [96].

In an experiment, Santello et al. [96] asked human subjects to shape their hand to grasp and use a number of objects. The imaginary objects were items of everyday use of different shape and size, which are normally held with a variety of power and precision grasps (e.g. "wrench": power grasp, "chalk": precision grasp). The hand joint angles were recorded using a 15-sensor CyberGlove, and principal component analysis (PCA, see Appendix A.1) revealed that human grasping is controlled in a low-dimensional subspace. In

fact, the first two principal components (PCs) shown in Figure 2.6 accounted for over 80% of variance in the recorded grasps, while the first three PCs explained 90% of the variance. The principle components are termed *postural synergies*, and the results indicate that the number of control parameters necessary for successful grasping may be significantly reduced. Although a small number of postural synergies accounted for a large part of the data, pairs of joint angles were poorly correlated in many instances. This suggests that the PCs of higher order do not simply represent noise in the system, but are contributing information about the hand shape corresponding the object.
Building on the concept of postural synergies, Bicchi et al. [8] present a model to introduce elasticity into the system. The *soft synergy model* combines two force fields to control the hand: An attracting field pulling the hand towards a virtual hand shaped according to the synergy manifold, and a field repelling the hand from penetrating the object. The dynamic equilibrium is found depending on the impedance of the actuation and control system. They show that this model is also able to predict grasp force distribution patterns. This work suggests the great potential of transferring the concept of postural synergies to artificial systems. Ciocarlie et al. [22] coined the term *eigengrasps* for the principal components and demonstrate how this concept can be used to reduce the dimensionality of the grasp planning problem in robotics. They optimize for grasps in the low-dimensional eigengrasp space and are able to find stable grasps for various robotic hands (e.g. Robonaut hand, Barret hand) and objects. Apart from aiding grasp planning for existing artificial hands, the postural synergy concept can also inspire the design and control of new robotic hands. Brown and Asada [10] for example present a mechanism that implements the hand synergies using a system of pulleys. Their mechanism and prototype represent one example of many *underactuated* hands developed by researchers over the last decades. The term *underactuation* describes systems with an input vector of lower dimension than the system output vector. In the robotic hands domain this refers to the number of actuators compared to degrees of freedom: Underactuated robotic hands have fewer actuators than degrees of freedom.

Many researchers have worked towards the understanding of the physical capabilities of the human hand. Similarly, much work has gone into identifying human strategies to synthesize and control grasps, and understanding transitions between grasps we perform in order to manipulate objects and execute desired tasks. Despite these efforts, human manipulation is still not well understood and state of the art robotic hands are still far behind humans in terms of dexterity. Nevertheless transferring human grasping and manipulation techniques is a promising approach towards the creation of more successful and widespread robotic hands.

2.3. Software Frameworks

The following sections briefly introduce communication between different computers via the Internet's Transmission Control Protocol and Internet Protocol (TCP/IP), and the Robot Operating System (ROS).

2.3.1. Network Communication

A computer network can generally be defined as a group of host computers which are connected together to execute a given task. In order to regulate communication within the network, a set of rules that govern the communication, a *protocol*, is necessary. A generic protocol model known as the Open System Interconnection (OSI) model characterizes and standardizes how different software and hardware components should divide tasks and interact in network communication. It defines a set of 7 different layers ranging from the physical layer, handling data as raw bits, to the top-level application layer handling the services presented to the user. Unlike the generic OSI model, the TCP/IP model was developed based on a set of existing protocols. A detailed description of protocols and different layers in protocol models is beyond the scope of this thesis and can be found in [63].

TCP/IP provides client-server based end-to-end connectivity through a network. Every device on a TCP/IP network is identified by its *IP address*, and the network *port* identifies the application or service running on the device. The interface which is used to send data through the network is termed *socket*, and a socket is associated with the local IP address and port number. On the server side, a socket is established and waits, "listens", for connection requests from a client. Once the client socket initiates the connection, bytestreams can be sent from the client to the server, and the server can process the received data.

To create socket applications, libraries and Application Programming Interfaces (APIs) are available for many operating systems and can be used in various programming languages, e.g. using the Winsock API in a C++ project.

2.3.2. Robot Operating System (ROS)

Robot Operating System (ROS) is a platform that supports the development of robot applications. Developed by Willow Garage[2], ROS provides a variety of libraries and tools to aid software development to control robots. The main ROS client libraries (C++: roscpp, Python: rospy) largely rely on open-source software dependencies, therefore the official ROS support is only provided for Linux (Ubuntu and Debian). For Windows, support is very limited and termed "experimental" (community based).

Basic components of ROS are:

Nodes: Processes that perform computation. Nodes communicate and exchange data with each other through the ROS communication interface.

Messages: Simple data structure of field types. Nodes can *publish* messages to topics, and *subscribe* to topics to receive messages that are published to the topic.

Topics: Named buses where nodes can exchange messages. The default transport method for sending and receiving messages is TCP/IP.

The most basic unit of the ROS framework is a **ROS package**, a functional module which can include nodes, additional libraries, configuration files and datasets. A variety of packages for various applications are publicly available online. Extensive information and tutorials on ROS can be found on the ROS Wiki[3] and ROS Answers[4].

2.4. Sensors

2.4.1. Vicon Motion Capture

Motion capture systems are capable of recording position and orientation of moving objects over time. These systems are used in order to animate human models in computer animation, but are also applied in various other fields, e.g. medical applications such as human gait analysis or validation in computer vision and robotics. Motion capture systems can be divided into *marker-based* and *markerless* systems.

The Vicon Motion Capture System[5] is a *marker-based* system which uses multiple high definition cameras to record trajectories of reflective markers attached to the test subject (or object). To obtain the 3D position of each marker, light is emitted from the Vicon cameras and reflected back from the markers. In contrast, *marker-less* systems such as the Microsoft Kinect sensor employ infrared or time-of-flight sensors to track motions without the need of markers.

Marker-based systems have been proven to be fairly accurate [34, 84] and are widely used to obtain validation data on motion.

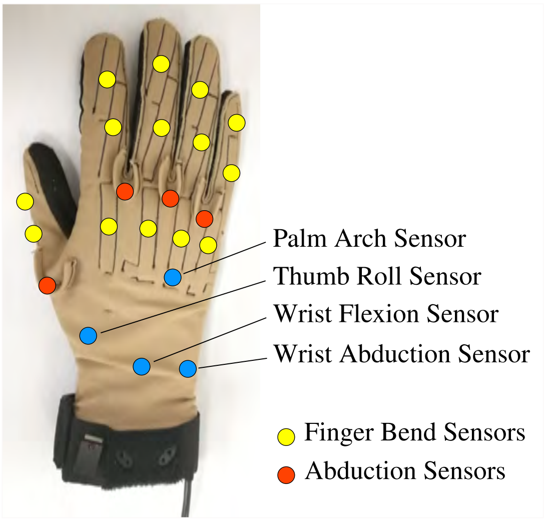


Figure 2.7.: CyberGlove sensor layout.

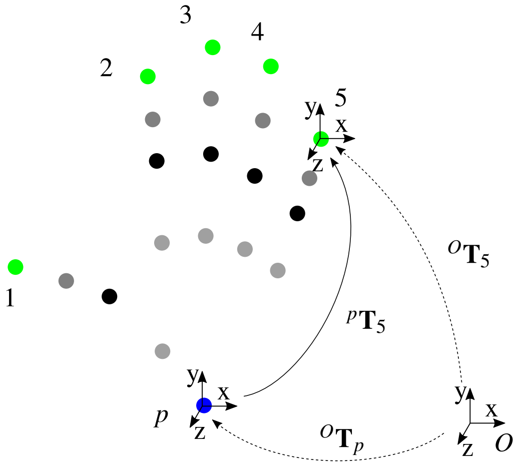


Figure 2.8.: Transformations of fingertip coordinate systems, illustrated for the pinky finger.

2.4.2. CyberGlove

Data gloves typically record hand posture by measuring the joint angles of the human hand. The data glove used in this work is a CyberGlove[6] which estimates hand posture using 22 resistive bend sensors with a resolution below 1 degree [28]. The sensor layout on the glove is shown in Figure 2.7. CyberGlove Systems provides a graphical user interface (the Device Configuration Utility) to manage and calibrate connected CyberGloves. The Virtual Hand Software Development Kit (SDK) includes a kinematic hand model and a set of C++ libraries to aid the integration of hand posture measurement into an application. The glove can be registered to the kinematic hand model, and the homogeneous transformations corresponding to the hand joints can be retrieved. With the CyberGlove SDK, the hand pose is obtained in the form of transformations relative to a global coordinate system O. To obtain fingertip poses invariant to wrist position and orientation, the palm base coordinate system is used as reference. The fingertip coordinate frames $^{p}\mathbf{T}_j$ for each fingertip $j = \{1,\ldots,5\}$ relative to the coordinate frame at the base of the palm p are obtained by inverting the transformation matrices of the palm base $^{O}\mathbf{T}_p$ given in the global coordinate system O:

$$\begin{aligned} ^{p}\mathbf{T}_j &=^{p} \mathbf{T}_O \, ^{O}\mathbf{T}_j \\ &= (^{O}\mathbf{T}_p)^{-1} \, ^{O}\mathbf{T}_j \end{aligned} \tag{2.4}$$

The transformations for the fingertip of the pinky are shown in Figure 2.8. Dots represent the 3d position of the hand joints, obtained with the CyberGlove. Green dots mark the positions of the fingertips.

2.5. Finite Element Simulation

The Finite Element Method (FEM) is a numerical approach to approximate behavior of continuous structures by minimizing an energy functional. FEM can be used to predict physical properties for a variety

[2] https://www.willowgarage.com/
[3] https://wiki.ros.org/
[4] https://answers.ros.org/
[5] https://www.vicon.com/
[6] http://www.cyberglovesystems.com/

of problems, ranging from displacements of mechanical structures to velocities of flows or temperature in heat conduction problems. A detailed description of general principles of FEM exceeds the scope of this work and can be found in literature [89].

According to Reddy [89], basic ideas of the FEM can be divided into the following steps:

1. **Finite element discretization:** The domain is divided into a finite number of *elements*, and these elements are connected with each other at *nodes*, forming a *finite element mesh*. The mesh resolution represent a trade-off between accuracy and computation time.

2. **Element equations:** The elements are considered isolated, and *element equations* approximating the desired properties are derived.

3. **Formulation of system of equations:** Element equations are assembled into overall system equations.

4. **Solution of the assembled problem:** The system equations are solved considering boundary conditions.

5. **Postprocessing of results:** Data of interest is extracted from the solution and can be visualized and further processed.

In this work, the FEM approach by Bern et al. [5], which uses FEM to predict deformation of plush toys, was drawn upon as foundation for the developed soft foam robot design and control tools. A more detailed description of how FEM is applied to predict deformations of soft foam bodies can be found in Section 6.1.

2.6. Machine Learning

Machine learning refers to the automated detection of patterns in data [102]. Over the last decades, technology has advanced rapidly, leading to an increasing availability of massive computing and storage resources. Machine learning has become a common tool in almost any domain where the extraction of information from data sets is required. A variety of toolboxes and tutorials on machine learning are available online, such as **scikit-learn**[7], an extensive Python module for machine learning. This section will explain basic concepts of machine learning techniques and algorithms that are used in this work. More detailed explanations can be found in literature [102, 9].

2.6.1. Supervised Learning

A *supervised* learning problem is characterized by the availability of training data consisting of a set of input values or vectors and the corresponding output values. In many applications, the goal is *generalization*, which can be explained as the ability to make a prediction about the output values for some new values of the input variables.

In contrast, if no corresponding output data is available for a set of input vectors in the training data, the learning problem is labeled *unsupervised*.

Supervised learning problems can further be categorized with respect to the nature of the output variable: If the output variable is described by set of discrete categories or labels, the term *classification* is used. *Regression* refers to a task with one or several continuous output variables.

[7] `scikit-learn.org`

Regression

Given a training data set consisting of N observations of the input variable $\mathbf{x}$, where $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_p \end{pmatrix}$ and corresponding observations of the target variable t. In *linear regression* the relationship between input and target variable is modeled as a linear combination of the p components of the input variable:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_p x_p \tag{2.5}$$

This model is fit to the training data set. A widely used approach is ordinary least squares, in which the sum of the squares of the error between prediction y_i for each data point $\mathbf{x_i}$ and the corresponding target value t_i is minimized:

$$\min_{\mathbf{w}} E = \min_{\mathbf{w}} \sum_{i=0}^{N} (y(\mathbf{x_i}, \mathbf{w}) - t_i)^2 \tag{2.6}$$

The term *multiple regression* generally describes models where multiple input variables and one output variable are available for each observation. Apart from linear regression, a further special case of multiple regression is *polynomial regression*, where the response variable is assumed to be a polynomial of the input features $\mathbf{x}$.

If a model has only "memorized" the training data and fails to generalize well for additional input data, it is called *overfitted*. A common approach to control and prevent overfitting is *regularization*, in which a penalty term is added to the error function E used in Equation (2.6). This penalty term discourages the coefficients in $\mathbf{w}$ from reaching large values:

$$\min_{\mathbf{w}} \sum_{i=0}^{N} (y(\mathbf{x_i}, \mathbf{w}) - t_i)^2 + \alpha \|\mathbf{w}\|_2^2 \tag{2.7}$$

In this case a quadratic regularizer is used (using the L2 norm), this is termed *ridge regression* [44]. The parameter α controls the relative importance of the regularization term compared to the sum-of-squares error term.

The *kernel method* is a method in which input data $\mathbf{x}$ is mapped into a feature space F prior to applying a learning or regression algorithm [103]:

$$\phi : \mathbf{x} \in \mathbb{R}^n \to \phi(\mathbf{x}) \in F \subseteq \mathbb{R}^n \tag{2.8}$$

The idea behind the kernel method is that nonlinear relationships can be mapped to a new feature space in which they are of linear nature and can therefore be captured by a linear regression model.

Neural Networks

An artificial neural network is a computation model inspired by the structure of the human brain, where a number of basic units, *neurons*, are connected with each other, forming a complex communication network.

A neuron is characterized by its *activation* function, a simple scalar function which is applied to the weighted sum of the neuron's inputs. One examples of an activation function is a rectified linear unit (ReLU) [79], which outputs the positive part of its argument, $f(x) = max(0, x)$.

Neural networks are usually described by a graph where nodes are the neurons and (directed) edges link the output of a neuron to the input of another neuron. *Feedforward* networks are structures wherein no cycles are contained in the underlying graph. Typically the network is organized in *layers*, where a number of layers, termed *hidden layers* can be placed between the input layer and the final layer of the network, the output layer. Networks with more than two layers are termed *deep networks*.

The goal of training a network is to find weights w, so that the network predicts the output values for a given input data set. The error can be calculated as the difference between predicted output and actual output, and the function to compute this error is termed *loss function*. During training, the error is minimized by propagating the error backwards, starting at the output layer, to a previous layer, to modify weights and bias. This is called *backpropagation*, and usually gradient descent is applied to modify the weights.

Recent studies have demonstrated that artificial neural networks can achieve cutting-edge performance on many learning tasks [102].

2.6.2. Reinforcement Learning

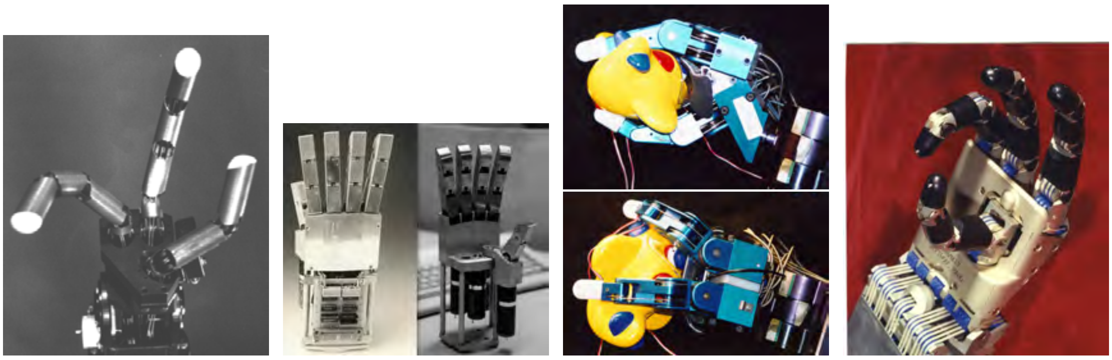
The term *reinforcement learning* [111, 110] refers to learning a mapping between situations and actions - which action to execute in a situation - in order to maximize a numerical reward. An major challenge in reinforcement learning is to find a balance between exploration of new actions, and exploitation of previous successful actions. As a complete description of methods, structures and algorithms used for reinforcement learning exceeds the scope of this work, for more information see Sutton and Barto [110], who give an extensive introduction to the topic.

3. Related Works

Over the years, soft robot hands and grippers have been designed from a wide range of materials and actuators. The following sections detail recent developments in the field, in particular in design and fabrication, and in modeling and control of soft robotic grippers and manipulators.

3.1. Robotic Hands

Inspired by the effortless human manipulation capabilities the robotics research community has always been interested in the development of multifingered robotic end effectors. In the late 1970s Okada [82] developed a tendon-driven three fingered hand which was capable of performing a nut-opening task. A pioneering design featuring an underactuation mechanism between the fingers is the Belgrade/USC hand [88, 4]. In the 1980s, two hand models developed by Stanford/Jet Propulsion Laboratory (JPL)[95] and the University of Utah/Massachusetts Institute of Technology (MIT) [51] both represented major breakthroughs in terms of hand designs [75].



a) Okada hand[1] b) USC/Belgrade hand[2] c) Stanford/JPL hand[3] d) Utah/MIT hand[4]

Figure 3.1.: Multifingered robotic hands.

Following these advances, numerous multifingered robotic manipulators have been developed by researcher centers all over the world, among them the Deutsches Zentrum für Luft- und Raumfahrt (DLR) hand(s) [14, 41], the NASA Robonaut Hand [69], the Karlsruhe hand [31], the Gifu Hand [59], the Barrett Hand [116], the Shadow Dexterous Hand [62] and many others.

Schulz et al. [100, 99] developed a hydraulically driven multifunctional prosthetic hand that can close approximate the human grasping abilities. This hand is driven by small sized flexible fluidic actuators [98] and has 15 degreed of freedom. Due to the small size and low weight of the actuators, they are integrated in the fingers of the hand, making the hand lightweight and compact. The fingers are self adapting, enabling robust grasping and more natural looking motions than conventional prosthesis.

Osswald et al. [83] explore an approach to integrate a hydraulically actuated hand into a humanoid robot, and present a high-level controller to connect low-level control of the hand with the rest of a

[1] Retrieved from `http://okada.eng.niigata-u.ac.jp/3finpict.jpg`

[2] Retrieved from `https://www.wired.com/images_blogs/photos/uncategorized/2008/09/26/usc_belgrade_hand.jpg`

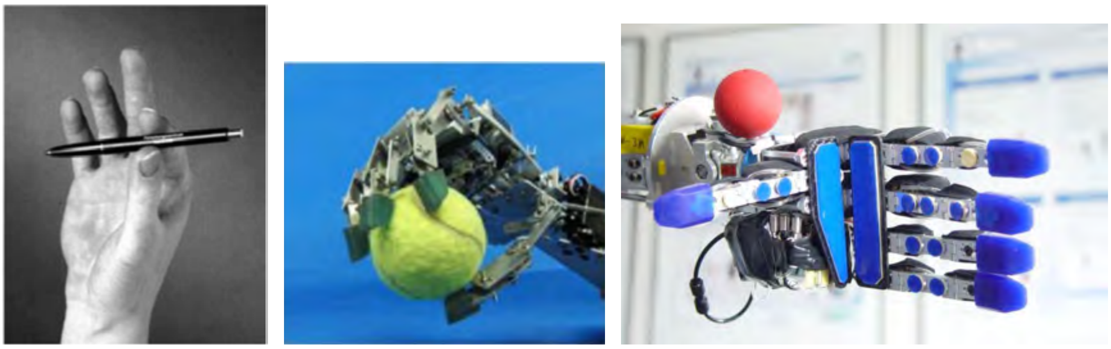
[3] Retrieved from `https://www.cc.gatech.edu/gvu/people/faculty/nancy.pollard/grasp.html`

[4] Retrieved from `http://www.computerhistory.org/collections/catalog/102693567`

humanoid robot.

Utilizing an underactuation mechanism, the TUAT/Karlsruhe Humanoid Hand [37, 38], automatically adjusts grasp shape and force without the need for sensors or feedback.

The hands of the humanoid robot ARMAR-III [2] are actuated with flexible fluidic actuators and equipped with position sensors on the joints to improve performance in exploration tasks and feedback [39]. This hand represents a hybrid between a five fingered anthropomorphic hand and a three fingered gripper.



a) Prosthetic hand. From [100].

b) TUAT/Karlsruhe Humanoid Hand. From [38].

c) FRH-4 Hand of ARMAR[5]

Figure 3.2.: Multifingered hands. Hydraulically driven prosthetic hand, underactuated hand, fluidic actuated ARMAR hand.

In applications where delicate handling of objects and safe interaction with humans is crucial, the use of compliant materials presents a promising approach, and a variety of research projects has been dedicated towards the design and control of soft robotic hands.

3.2. Design and Fabrication of Soft Robots

3.2.1. Materials and Actuation

In general, the rigidity of materials can be described using Young's Modulus. While rigid robots are usually composed of materials with moduli in the range of $10^9 - 10^{12}$ pascals, the rigidity of biological systems (e.g. skin, muscle tissue) is significantly lower ($10^4 - 10^9$) [94]. Rus and Tolley [94] define soft robots to be "(...) primarily composed of materials with moduli in the range of that of soft biological materials". In this work, the term soft robots strictly refers to robots that are build of soft materials

Recently, Shintake et al. [106] have reviewed existing soft robotic grasping technologies extensively. Their work divides gripping technologies into three main categories:

- **Gripping by Actuation** relies on gripper fingers or elements which bend around the object and are either actively controlled or exploit passive adaption to object shapes. Since the gripping mechanism in this thesis is actuation based, the following review will focus mainly on this category.
- **Gripping by Controlled Stiffness** uses large changes in material rigidity to grasp and hold objects. Materials used in these types of soft grippers are granular materials such as coffee grounds [11], which are embedded in an elastic bag. The stiffness of the gripper is varied using vacuum pressure control enabling the gripper to grasp and hold a variety of objects.
- **Gripping by Controlled Adhesion** is implemented by regulating the interface attraction between gripper and object surface.

[5] Retrieved from `http://grasp.xief.net/data/images/armarhand.jpg`

The most widely used materials for soft robots and particularly soft robotic grippers are elastomers [106, 115, 67, 21, 104] such as silicone rubber. They are easy to fabricate and biocompatible [61], a property allowing their use in direct contact with humans, such as in medical applications.

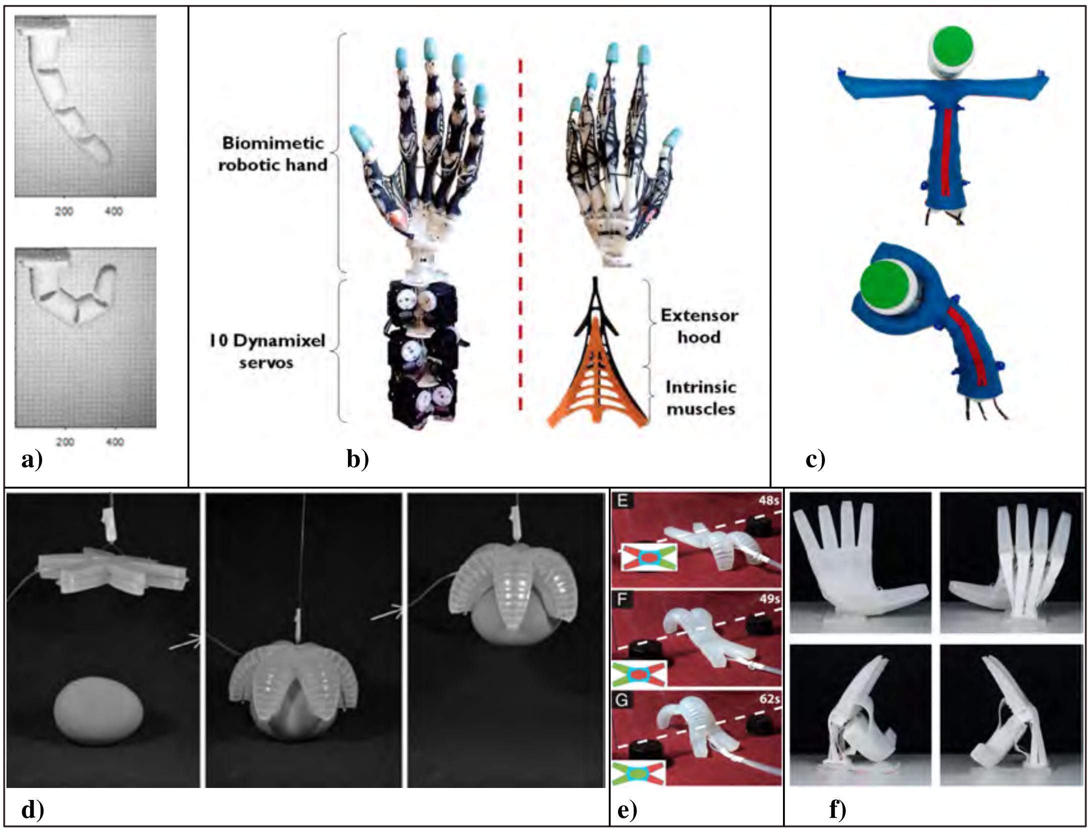


Figure 3.3.: Soft hands. Tendon-driven: a) 3D printed finger [78], b) biomimetic anthropomorphic hand [119], c) plush 2D gripper [6].
Fluidic actuated: d) starfish gripper [49], e) quadrupedal robot [104], f) RBO 2 Hand [29].

Actuation of gripper or hand segments can be implemented using a variety of different approaches.

Tendons

In tendon-driven hands, the segments are passive structures, which are controlled by variable length tendons driven by external motors [94]. While conventional tendon-driven systems are build from rigid links and joints, a promising approach is to replace some or all of the rigid structures in these systems with soft materials. Mutlu et al. [78] for example present a monolithic 3D printed finger composed of thermoplastic elastomer which is actuated by tendons routed through internal channels and demonstrates full compliance and adaptivity to object shapes.

Xu and Todorov [119] have developed an anthropomorphic hand design which is tendon-actuated and closely mimics human bone geometry and tendon and muscle placement. They are able to demonstrate most of the grasp types defined by the Cutkosky taxonomy (Section 2.2.3). Since it contains rigid structures this hand can however not be considered entirely soft. Furthermore, it is a highly complex system, making fabrication and control of the hand hardly accessible for non-experts. Bern et al. [6] present a class of tendon-driven plush robots. The bodies of these robots are build from a textile skin filled with standard polyester fiber filling and tendons are sewn through the skin and driven by servo motors. However, these robots are intended as toys for children, and are not suitable for dexterous manipulation.

Fluidic Elastomer Actuators

A further method to achieve desired deformations of gripper segments is to inflate or pressurize channels embedded into the soft structure. Despite being among the oldest technologies, it is until today the most widespread technology to achieve soft robotic actuation. Pneumatic Artificial Muscles (PAMs, also termed McKibben actuators) are elastomer tubes in fiber sleeves [20] and have been used as muscle-like linear actuators, often in an antagonistic setup.

Fluidic Elastomer Actuators (FEA) are actuators composed of elastomer chambers which expand when pressure is exerted by a fluid [106]. FEAs are operated pneumatically or hydraulically and are controlled by relatively low pressure. Ilievski and his colleagues [49] use embedded pneumatic networks (PneuNets) of channels in elastomers to build a starfish-like gripper, achieving complex motions with only a single pressure source. PneuNets are also used in a quadrupedal robot capable of locomotion [104].

The anthropomorphic RBO Hand 2 developed by Deimel and Brock [29] is made of silicone rubber and uses FEAs enforced with polyester fibers. Their PneuFlex actuators can be manufactured rapidly and uses cheap and non-toxic materials. The RBO 2 Hand can robustly grasp objects of various shapes, its performance is evaluated using the GRASP taxonomy [36]. Further designs actuated with FEAs include silicone polymers and elastomers combined with different materials such as paper or cloth [94].

FEAs are easy to fabricate, robust and use low-cost elastomers. However, the need for a compressor and pressure-regulating components as well as the risk of failure of the overall system resulting from leakage limit their scope of application.

Apart from pneumatic and hydraulic actuation, electrically activated actuators composed of electroactive polymers (EAPs) have been used to build robotic grippers, but exhibit a relatively slow actuator response and produce low stresses [106]. A further actuation approach is shape memory materials: They incorporate polymers or alloys which after being deformed return to their initial shape in response to a (usually thermal) stimulus. The application of these systems is restricted by shape recovery speed and hysteresis effects.

3.2.2. Design and Fabrication

Soft robotic systems are usually created using conventional 3D computer aided design (CAD) software [94]. These tools are however in general not intended or easy to use for the design of free-form shapes and complex non-homogeneous systems. As a result, researchers use either 2.5D layered designs, or develop their own custom design and fabrication tools.

A variety of designs are created either based on human intuition, drawing inspiration from biology or by evaluation of anthropomorphic anatomical data. Following a different approach, a number of automated design processes that optimize designs of soft robots have been proposed. Hiller and Lipson [43] developed a custom finite element analysis (VoxCAD), which is used to generate and optimize the design of a soft locomotion robot using an evolutionary algorithm. Evolutionary algorithms are also used by Rieffel et al. [90] to find optimal designs and locomotive gaits for a soft robot. Deimel et al. [30] optimize morphology and control signals simultaneously for a soft hand, but limit the problem setting to a simple grasping scenario and a fixed general morphology with variable segment lengths. Inouye and Valero-Cuevas [50] evaluate the grasp quality of a precision grasp using an anthropomorphic kinematic layout. This hand is not soft, but their work demonstrates the great potential of optimized hand designs. They show that optimization techniques are capable of significantly increasing grasp quality of hand designs, even exceeding human performance.

A powerful fabrication method widely used for rapid prototyping and custom shapes is 3D printing. To fabricate soft robots, 3D printing is used to create negative or positive molds for casting processes [67] or robot parts, soft or rigid, are printed directly from digital designs [78]. A further efficient and easy technique to create 3D shapes is casting molds from materials such as silicone rubber [104, 115, 67] or flexible foam. Shape deposition manufacturing (SDM) applies alternating cycles of material deposition and shaping and can be used to create shapes and plant structures inside them. Cham et al. [16] use

SDM to create hexapedal robots and embed servos and wiring inside the robot's body. To fabricate fluidic actuators, in addition to SDM, soft lithography [118] represents a method to create internal channels and chambers for actuation. Using this technique complex patterns on polymer materials can be achieved.

3.3. Modeling and Control Strategies for Soft Robots

Unlike rigid robots, the movements of soft bodies cannot be described by six degrees of freedom. Elastic materials show effects such as buckling, twisting, stretching, compression and wrinkles. Their motion can be viewed as having an infinite number of degrees of freedom. Furthermore soft robots are fabricated from a variety of different materials and actuators. The high compliance and the wide range of design and actuation techniques makes modeling and controlling soft robots a difficult task, and requires new strategies for modeling, control, dynamics and high-level planning [94].

3.3.1. Modeling

A common approach to model the kinematics of soft robots is to use a simplified model which assumes piecewise constant curvature (PCC) [94]. This model has been implemented using different techniques (e.g. Euler-Bernoulli beam mechanics) [117, 54]. As they integrate robot morphology and actuation characteristics, these models are highly specific to each robot. Once the model is established, it provides a mapping from the actuation space to the configuration space and represents a simulation model that predicts solutions to the forward kinematics problem (Section 2.1.2). The constant curvature model has been shown to be a good approximation for uniformly shaped manipulators with symmetric actuation design with minimal torsional and external loading effects [114].

A further modeling approach for more complex shapes is to create a discretized model of the soft robot and use the finite element method (FEM) to predict deformations resulting from actuation [78, 43, 49]. The SOFA framework[6] is dedicated to the development of physics-based simulations. A soft robots plugin for this framework has been developed by Coevoet et al. [24] dedicated to the modeling, simulation and control of soft robots. For a cable-driven trunk-like robot fabricated from silicone, they compare trajectories of the real robot with their simulation models, with an average error of 4.72 mm. The current implementation assumes linear elasticity which restricts deformations of the material to be small, therefore large deformations cannot be handled.

The finite element modeling approach used to simulate plush robot behavior [6, 5] is drawn upon in the soft foam robot modeling software developed in this thesis. A more detailed explanation of the FEM approach used to model foam robots can be found in Chapter 6.

3.3.2. Control

Similarly to materials and actuators, researchers have often drawn inspiration from biological systems to derive control strategies for soft robotic systems. As an example, the gait patterns of caterpillars have been used as an ideal model to control soft robotic locomotion [67], and the precise movements of the octopus have been adapted to control an octopus arm robot [73].

Thuruthel et al. [114] have recently reviewed control strategies for soft robots and identify three categories of controllers: Model-based, model-free and hybrid controllers (a combination of both). Previous works have particularly studied the problem of inverse kinematics (IK) which is concerned with finding a mapping between actuation and desired configuration.

Model-Based Controllers for Inverse Kinematics

The most widely used and studied approach for soft robots are model-based controllers. One of the most reliable methods is the use of the constant curvature (CC) assumption and model, and the inversion of this model to predict actuations. Compared to more complex models, the CC approximation allows

[6] www.sofa-framework.org

relatively fast computation and is not design specific. Since more complex methodologies are computationally expensive and require the estimation of many parameters, they have not achieved exceptional improvements in performance [114]. As an example of model-based control, Saunders et al. [97] model caterpillar-like soft robots as a series of extensible linkages. For tentacle-shaped soft robots Marchese et al. [72, 71] and Chen et al. [19] use PCC models to approximate the robot. For soft robots with arbitrary shapes work by Duriez [32] presents a real-time solution using a finite element method (FEM).

Recent advances in computation and storage capacities promote data-driven controllers as an alternative approach to solve the inverse kinematics problem.

Model-Free Controllers for Inverse Kinematics

Model-free approaches require the collection of data samples of actuations and corresponding robot configurations, which then serve as training data for a learning method. These data samples are obtained either using a simulation framework, where given an actuation pattern the corresponding robot configuration can be predicted, or are collected through sensor measurements. Both approaches suffer from inaccuracies. Simulated samples underly uncertainties due to incomplete or incorrect modeling of the robot system while sensor-based sampling is affected by sensory noise and measurement uncertainty.

Neural networks have successfully been used to learn inverse kinematics on a cable driven soft tentacle manipulator with 2 degrees of freedom [40]. Rolf and Steil [92] have proposed an efficient exploration algorithm for creating task space samples for IK learning.

Since they enable the control of soft robots with arbitrary complex kinematics and circumvent the need to create exact analytical models of the robot, model-free approaches represent a promising method for highly nonlinear, non-uniform systems, which can be influenced by gravity and/or act in uncertain environments [114].

Teleoperation

Teleoperation generally refers to robotic systems with a human operator in control, or *human-in-the-loop* [107]. First developed for safe handling of radioactive material and space operations, teleoperation is nowadays used in a variety of applications. An example for a successful medical application is the da Vinci telesurgical system [42].

Teleoperation of robotic hands is usually achieved by recording the hand configuration of a human operator, and a mapping process which determines the necessary hand actuation to reproduce the desired configuration with the robotic manipulator. For grasping or manipulation tasks the human operator can visually guide the robotic hand motion and represents a high-level controller. Xu and Todorov [119] for example use a custom data glove to record human hand configurations and operate a biomimetic anthropomorphic robotic hand.

For soft manipulators, specifically non-anthropomorphic designs, teleoperation is a non-trivial task. This is due to significant differences between the human and robot hand with respect to geometry, as well as deformation behavior. In order to teleoperate soft and non-anthropomorphic manipluators, it is therefore necessary to learn a mapping from user gestures or poses to motor actuations that deform the robot in the desired manner. Inspiration can be drawn from research on puppeteering in computer graphics. For example, Seol and colleagues [101] present a method that allows the user to specify how they wish to move in order to create certain character motions. As an example, they might choose to swing their arm to move an elephant's trunk. In the case of [101], an approach based on feature mapping is used to convert from user motion to character control parameters.

3.4. Challenges and Contributions

Although a large body of research works has been dedicated to developing soft robotic hands, there is still no system available that allows easy and straightforward design, accessible fabrication and intuitive control. To create new models of soft manipulators, specialized design tools to create task-specific hands are hardly available and require expertise (e.g. to design shapes and layout of fluidic actuators). Using optimization techniques to automatically derive designs has been investigated for locomotion [43, 90] and grasping [30, 50]. However, these approaches do not capture the fully unbounded and complex nature of grasping and manipulation, but optimize grasping within a constrained design domain (e.g. restricted shapes, planar scenarios). Once a digital model of the design is created, fabrication and assembly of soft manipulators are often tied to complicated or expensive manufacturing techniques.

Successful grasping and manipulation requires dealing with uncertainty and cluttered environments. To achieve this goal, a variety of control strategies for robotic hands have been explored in simulation, but are yet to be transferred to physical systems. Especially traditional robotic manipulators have difficulty to interact with uncertain environments due to their rigid links and joints.

In contrast, humans exploit the compliance of their hands to perform daily tasks. Entirely soft robots are therefore a promising approach to achieve robust grasping, and they may even make precise low-level control strategies unnecessary as they can instead rely on their structural compliance. Still, intuitive control methods are necessary to successfully perform grasps in real-world scenarios.

Many research groups focus on improving specific abilities of highly complex systems, while easily fabricated and yet dexterous soft manipulators remain largely out of reach. As long as soft robots are not inexpensive and available to the novice user their acceptance will not increase, preventing their widespread use in our daily lives.

To help democratize the process of creating and controlling truly soft and dexterous manipulators, this thesis makes the following contributions:

- **Overall system:** The overall system of tendon-driven soft foam robot hands is created, consisting of various tools and methods to easily and rapidly create and operate such hands.
- **Fabrication:** A set of methods and procedures to fabricate foam hands, which can be extended to a variety of shapes.
- **Interactive Design and Simulation Tools:** A simulation framework which enables users to rapidly prototype foam robots of arbitrary shape and intuitively test motion capabilities.
- **ROS-based Control Framework:** A modular software package to control foam robots.
- **Learning IK on Physical Robot:** A learning-based approach to solve the inverse kinematics of a foam hand, which uses data samples obtained from a physical foam hand robot.
- **Teleoperation:** An intuitive approach to control foam hands using a CyberGlove.

4. Soft Foam Robot Hands

Tendon-driven soft foam robot hands are built of a foam core, covered by a textile skin. They are actuated by servo motors which pull on tendons that are sewn through the textile skin.

The development of these robot hands is motivated by the need for a lightweight and low cost soft robotic hand, which is easily built and controlled. A major goal for these robots is therefore that their design, fabrication and control should be accessible in terms of complexity, cost efficiency and intuitive operation. To promote this accessibility, users have to be provided with easy and straightforward tools to create and control task-specific robot hands.

In summary, the goal of this work is to consider the domains of fabrication, design and operation, and to develop, test and evaluate the necessary software, methods and processes to easily and rapidly create and operate this new type of soft foam robot hands.

Fabrication

The original fabrication process (described in Section 5.1) to create foam hands is a manual procedure, requiring a multitude of different casting steps and limiting hand designs to foam recreations of physical objects. In Section 5.2, this fabrication process is extended and improved to enable the fabrication of arbitrarily shaped hands and facilitate the procedure.

Design

Before a foam hand is fabricated, a user must decide on the design of the manipulator. The design of a hand is determined by the **hand morphology**, which is characterized by the number and shape (length, thickness) of the fingers, and the shape of the palm. A further aspect is the **rest pose** of a foam hand: This is the pose the urethane foam compound is cast into (e.g. a flat, stretched-out pose, or a pose where the thumb is opposing the fingers). A foam hand design is additionally specified by the **tendon routing**, which determines the number of tendons that are used and their placement on the hand.

Traditionally, robotic hands are manually designed from scratch by experts. However, since foam hands are composed of entirely soft materials, it is not easy and intuitive to predict the deformation of a hand resulting from tendon actuation. Achievable poses and motions depend on the hand design (morphology, rest pose, tendon routing). It is therefore desirable to have tools to investigate and analyze a hand design before the physical prototype is built, and thereby create designs tailored to the desired task.

To explore motion capabilities of a hand design and shorten design iteration cycles, an interactive simulation framework is developed, validated and applied in Chapter 6.

Operation

To be able to grasp and manipulate objects with a foam robot hand, appropriate control mechanisms are necessary. Apart from being efficient in terms of speed, accuracy and computational resources, a hand control method should additionally be intuitive, so a user can easily operate the hand. In Chapter 7, a modular control framework based on ROS is developed and an intuitive teleoperation method based on fingertip positions is explored. Different learning based techniques to solve the inverse kinematics of foam hands are furthermore tested and compared in simulation.

5. Fabrication of Soft Foam Robot Hands

This chapter presents a set of fabrication techniques and mechanisms to create a soft foam robot hand, which can be adapted for the creation of a wide variety of foam robots. Section 5.1 describes the manual fabrication process, which was established previous to this work. A set of methods and procedures are proposed to improve and extend this fabrication process (Section 5.2), and effects of different tendon routings and rest poses of a foam hand are explained (Section 5.3)

5.1. Manufacturing Foam Robot Hands

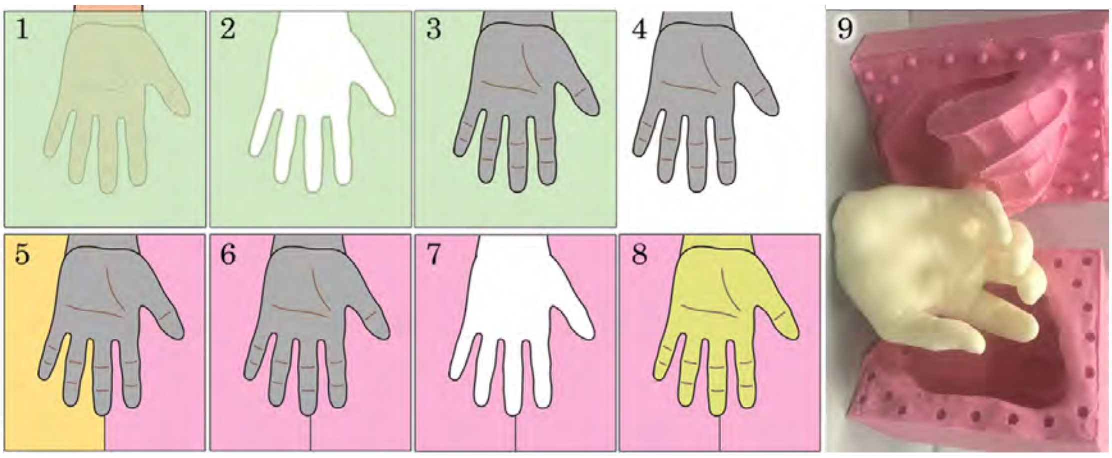


Figure 5.1.: Fabrication Process: 1) Cast of human hand in alginate, 2) Remove hand from alginate mold, 3) Pour plaster of paris into alginate mold 4) Carefully remove plaster of paris cast from alginate mold, 5) Cover one half of plaster model with clay and cast the other half from silicone, 6) Remove clay and fill second half with silicone, 7) Remove plaster model from two-part silicone mold and clean mold, 8) Cast foam hands using the master two-part silicone mold, 9) Final result. *Image by Jonathan King.*

The original fabrication process shown in Figure 5.1 was already established by the beginning of this thesis work, and a first prototype shown in Figure 5.2 was already built. This thesis work addresses several limitations of the first prototype, Section 5.2 details the extensions and improvements that were applied to the fabrication process and the physical robot.

The general goal of this type of robots is to achieve ’true’ softness, while maintaining the ability to perform a diverse set of tasks. A foam robot dexterous manipulator which is capable of performing complex poses and actions is therefore an ideal candidate to demonstrate the potential of this class of foam robot systems. To be ‘truly’ soft, the robot is constructed of only soft materials: foams, knitted textile skins, fibrous tendons, and flexible PTFE tubes for cable routing. Rigid mechanical components are housed away from the hand, and future systems could embed these structures inside the foam to a degree that their hardness is not noticeable. A further approach is to even replace all rigid structures by soft actuators (e.g. cost effective artificial muscles [81]).

Apart from building soft hands, the primary goal of this work is to develop a fabrication methodology which is easily accessible to non-experts. This requires the fabrication methodology to be low-cost, and to rely on easy to follow casting techniques for which step by step instructions can readily be found in online video tutorials. The mechanical parts used for the robot are off-the-shelf components, 3D-printed parts, and laser cut acrylic, and are therefore accessible to the novice user.

Creating the Mold

An initial hand pose, referred to as *hand rest pose* is determined by user intuition. To cast an anthropomorphic hand model, the mold is created using well-known silicone “life-casting” techniques for which the process is shown in Figure 5.1:

1.-2. A negative of the hand model is created by keeping the user’s hand in a bath of alginate. Alginate sets in less than 5 minutes, minimizing the time the user has to hold still, and is safe for continuous exposure on human skin.

3.-4. Plaster of paris is poured into the alginate mold to create a positive hand model.

5. Once the plaster model is removed from the mold, clay is manually applied to the plaster model, in order to cover one half of the model in clay to create a two-part silicone mold. Depending on the hand shape, this process can take several hours and requires careful design of the parting line of the two-part mold, in order to guarantee that positive hand models can later be easily removed from the mold. Silicone is poured onto the model to create the first piece of the two-piece mold.

6.-7. Once the silicone has set, the previously applied clay is removed, and the second piece of the silicone mold can be cast. This is the final mold creation step, and the two-piece silicone mold can now be used to cast foam hands in it.

8.-9. The foam core in the shape of a hand is cast using the master two-part silicone mold.

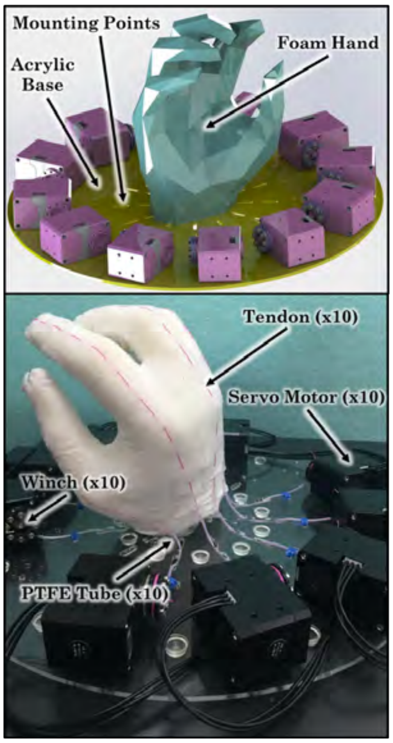


Figure 5.2.: Annotated assembly of the first foam robot prototype. *Image by Jonathan King.*

Casting Foam Hands

After applying a mold-release agent to the mold, the final foam hand is cast using a two-part urethane foam compound[1]. A two-part silicone mold and the foam hand cast in it are shown in Figure 5.1 9. Urethane foam is available in a variety of densities to choose from depending on the user’s application. For the foam hands in this work, FlexFoam-iT! X was used as it represents a good trade-off between strength and compliance. A close examination of the foam hands produced by this process showed that the deformation behavior of the foam is not always intuitive. As an example, slightly thicker sections of the palm are much stiffer, while slightly thinner sections of the fingers are much softer. To achieve very consistent results from cast to cast, a laboratory mixing machine is helpful, but not absolutely necessary.

Once the foam hand is fabricated, a simulation model of the physical hand can be obtained by 3D reconstruction using Autodesk ReMake[2] to generate a surface mesh from approximately 50 images of the hand taken with a smartphone.

The described casting process enables the creation of foam hands from a human hand in a chosen rest pose, while the cost of the mold is approximately $50, and only a few dollars for each foam cast afterwards. However, many tedious and time consuming casting steps are required, and the described method can only recreate positive models of hands. A time and cost efficient fabrication process is an important motivation for more researchers

[1] Smooth-On FlexFoam-iT! Series

[2] `https://www.autodesk.com/products/remake/overview`

to further experiment with this type of robots. Additionally, it is not intuitive to predict deformation behavior of a foam hand. To avoid unwanted behavior, the desired hand geometry should be carefully considered.

Gloves and Sewing Tendons

After the foam core is cast, a textile glove is fit and laminated onto it using spray-on upholstery adhesive. Tendons for actuation can then be sewn into the glove in arbitrary patterns. For anthropomorphic hand shapes, off-the-shelf gloves can be used, and for general soft robots, sewn skins from cut felt can be utilized. As a further alternative, textile skins can be custom knit by automatic processes [74]. This work enables custom gloves to be knit in under an hour and lets the user choose from various materials. PTFE coated braided fishing line serves as tendons, and the tendons are sewn into the glove with a typical sewing needle, and fixed at the ends with finishing knots.

Robot Chassis

The gloved hand is attached to a laser-cut acrylic base using hot-melt glue. To minimize friction, the tendons are routed through PTFE tubes and connected to servo driven winches. The PTFE tubes can be fixed with cable ties at additional mounting points on the acrylic base. The assembled robot is shown in Figure 5.2.

Since the hand is glued to the platform, this design does not allow to switch between hand prototypes without ripping the hand off the acrylic plate and thereby damaging or destroying the foam core. A further limitation is caused by the routing of tendons from the hand to the motors: The motors are placed in a perpendicular plane with respect to the hand, leading to a significant redirection of tendon forces, in this case from a vertical to a horizontal direction. This causes detachment of the glued glove from the foam body and of the foam body from the acrylic plate.

5.2. Extension and Improvements

While the first foam robot hand prototype was already capable of complex posing and actions (shown in Figure 7.7 left and Figure 7.8 top), several issues had to be addressed to improve its performance. The following sections each first describe a limitation of the original design and then explain the measures that were implemented to resolve the issues.

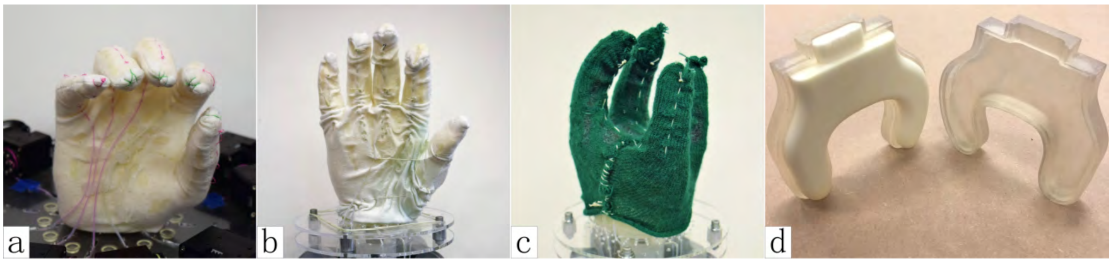


Figure 5.3.: Foam manipulator rest poses. a (first prototype) and b (flat hand) are fabricated by taking casts of human hands, while c (four fingered hand) and d (mold and foam cast of planar gripper) are non-anthropomorphic designs fabricated from digital 3D models.

Mold Fabrication

The fabrication process described in the previous section requires a physical hand model (a human hand), and involves several casting steps (alginate - plaster - silicone).

To fabricate foam bodies of arbitrary shapes, a 3D model of the desired geometry is required. This model can be obtained by either 3D scanning the desired physical model, or using software to create 3D models.

Since conventional CAD software does not provide easy and intuitive design tools of freeform shapes, *SculptGL*[3], a free online sculpting tool, is used to create a 3D model of a non-anthropomorphic four-fingered gripper, which is utilized to create the foam hand shown in Figure 5.3 c.

To reduce the necessary fabrication steps, either CAD software or automatic mold generation methods [120, 66] can be used to directly create mold models that can be machined or 3D-printed, using filament-based printers or stereolithography. This technique is applied to create the mold depicted in Figure 5.3 d, which is used to cast a planar gripper.

For general foam robots, if a digital 3D model is available, re-usable 3D-printed molds represent a rapid and cost-effective fabrication method to create the foam body.

Platform Design

The design of the acrylic platform, shown in Figure 5.2 promotes a detachment of the glued glove from the foam hand, and of the foam hand from the platform. This is caused by the perpendicular placement of the motors, which leads to the tendons being pulled more towards the side than in a straight direction.

Therefore a new platform is designed using CAD software. Similar to the previous design, the platform components are off-the-shelf parts (screws, bolts and nuts) and laser-cut acrylic. Tendons are routed straight from the hand through routing points in the platform to prevent the detachment of glued parts resulting from actuating tendons in a perpendicular direction. The new platform is shown in Figure 5.4, with the sewn part of an exemplary tendon actuating the thumb marked with a dashed blue line. The tendon is routed through a PTFE tube, highlighted in orange, and interfaces with a winch, which is attached to a servo motor. In this new design, the top plate can easily be removed and replaced. This allows to switch quickly between different foam hands to test different prototypes, as shown in Figure 5.5. The new platform design further allows for arbitrary tendon connection points and therefore enables more complex routings.

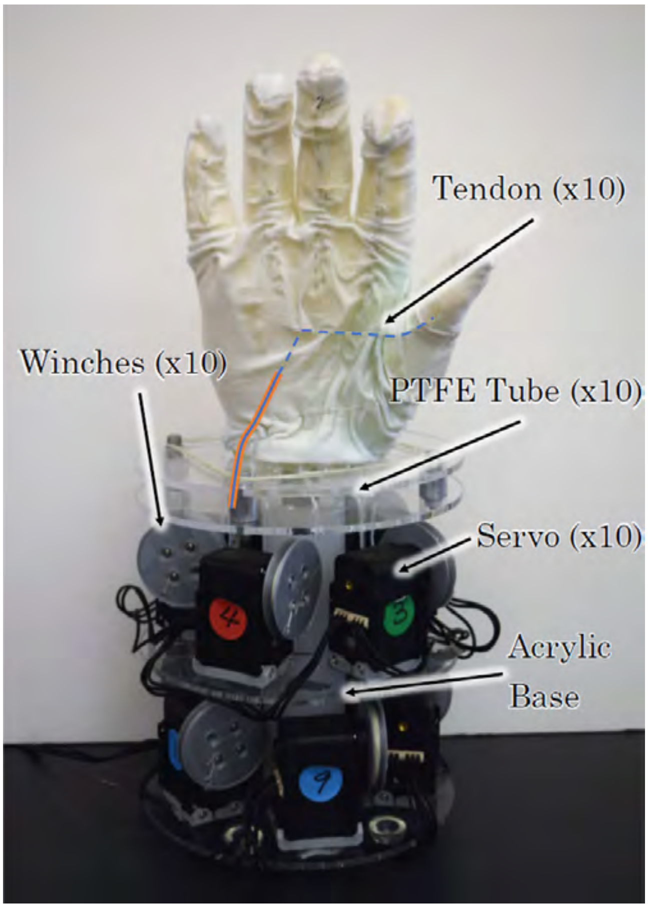


Figure 5.4.: Annotated assembly of a foam hand. A single tendon is highlighted in blue, a dash line represents the portion sewn into the glove. The corresponding PTFE tube is highlighted in orange.

[3] https://stephaneginier.com/sculptgl/

Interactive and Iterative Design in Simulation

A digital representation of the hand model, which is required to explore tendon routings in simulation, is only available from 3D scanning the physical, fabricated hand. Additionally, no easy and intuitive software is available to create and test tendon routings for a hand model rapidly and predict deformations and motion capabilities of a hand model.

In order to enable rapid design cycles, software was developed to explore hand deformations and tendon routings in simulation, without the need to first build a physical prototype. The developed simulation modules are described in Section 6.2.

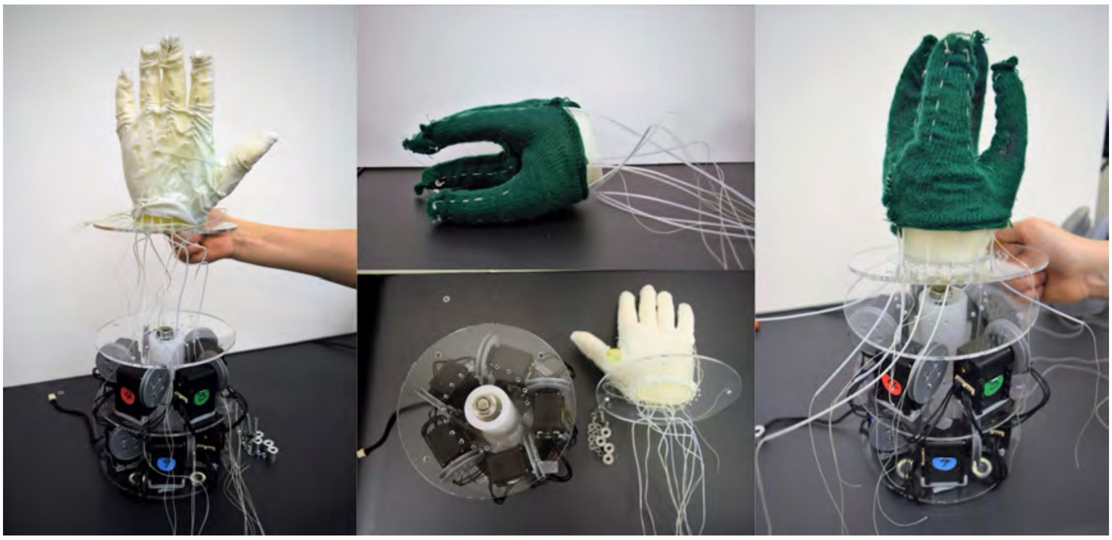


Figure 5.5.: Due to the modular design of the robot platform, foam gripper prototypes can easily be switched to allow rapid testing of prototypes. An anthropomorphic design is removed from the platform (left) and replaced by a four-fingered gripper (right).

5.3. Effects of Different Tendon Routings and Rest Pose

Since foam robots consist of a continuous structure instead of the links and joints of traditional robots, the variety and complexity of achievable poses largely depends on the **tendon routing** and the **rest pose**. Different rest poses and the effects of different routings can be explored in simulation (Chapter 6), before they are applied to a physical hand.

Tendon Routing

A major weakness of the design of the initial prototype (Figure 5.3 a) in terms of motion capability is the inability of the thumb to abduct and oppose the palm. This is primarily caused by an inefficient tendon routing with two antagonistic tendons as shown in Figure 5.6 on the left.

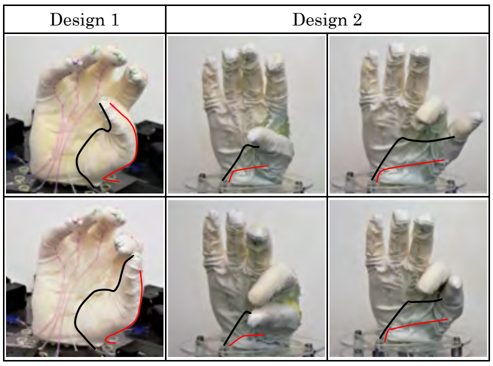


Figure 5.6.: The tendon configuration impacts the realizable poses of the foam robot hand. Left: Initial design, flexor and extensor. Right: Revised tendon routing, improving thumb mobility.

For a second prototype, the routing was changed according to the right column of Figure 5.6, which increased the complexity of feasible motions of the thumb significantly. With the revised routing, different tendon activations enable either lateral or opposing grasps as shown in Figure 5.6.

Rest Pose

The shape of a foam robot hand is determined by the **hand morphology** and the **rest pose**. A hand morphology is composed of the number of fingers, lengths of individual fingers, thickness of fingers and the shape and thickness of the palm. It is important to consider the aspects of this morphology: Relatively thick palms and fingers may lead to (possibly undesired) localized stiffening. Undesired features of a hand morphology can be mitigated by iteratively changing the hand designs and testing them in simulation, which reduces the need for iteration of the physical prototypes.

Once the morphology is determined for a foam robot, a further design aspect is the rest pose the foam is cast into, e.g. a cupped rest pose where the fingers are slightly curled, or a flat hand rest pose where the fingers are stretched out. Designs with different morphologies and rest poses are shown in Figure 5.3.

Observing the physical hand moving and posing, it becomes apparent that additional to hand morphology, the rest pose of the hand design clearly pre-defines the range of motion independently of the tendon arrangement. As an example, for a flat hand pose it is difficult to achieve poses where the fingers oppose the palm, whereas a cupped rest pose is already close to the desired pose and requires only small tendon contractions to achieve such a pose. On the other hand, for a cupped rest pose it is not possible to fully stretch out fingers into a flat hand, because independent of the tendon routing, fingers cannot be lengthened by contracting tendons.

Unlike the tendons, the hand rest pose is fixed and cannot be changed after fabrication, it is therefore important to evaluate rest pose geometries in simulation before fabricating the actual foam model.

Once a hand morphology is chosen for a hand, it depends on the underlying task which rest pose is suitable. This particularly applies to anthropomorphic hand geometries. From the fabricated prototypes, one observation is that human-like hands with flat rest poses have a problem grasping large objects such as a tennis ball. The fingers are not able to curl around the object and oppose the palm. However an advantage of flat rest poses over curled rest poses is that they do not require extensor tendons running on the back of the hand. Instead, due to the geometry and the elasticity of the material itself, once tension is released from a flexor tendon, the compliant material restores the hand's original pose. Without the need of extensor tendons, more tendons can be added to the front of the hand to enable more complex motions.

6. Interactive Design and Simulation Framework

Large elastic deformations and infinite number of possible tendon routings make it difficult to predict the behavior of designs without the support of design tools. Approaches that can automatically create efficient designs based on functional goals are still a long way from being able to address to full complexity of all design domains [3]. Therefore the following section describes methods and tools that support the manual design of soft foam hands.

6.1. Modeling Hand Deformation

To predict deformations of foam hands resulting from tendon contraction, an existing simulation framework which implements a finite element model was extended to interactively simulate foam hands. This framework is based on the work of Bern et al. [5], who animate and design tendon-actuated plushies. Adapting their work, the contractile elements are modeled as unilateral stiff springs, and the deformed, statically stable (minimum energy state) configuration of the model is referred to as $\mathbf{x}$. In $\mathbf{x}$, the 3D coordinates of all nodes in the simulation mesh are stored. $\mathbf{X}$ describes the undeformed configuration, also termed rest configuration. The total deformation energy of the system is described by

$$E = E_{foam} + E_{contractile} + E_{pins} \tag{6.1}$$

with E_{foam} denoting the energy resulting from the deformation of the simulation mesh, $E_{contractile}$ is the strain energy of contractile elements, and E_{pins} models the behavior of stiff springs, by which a small number of simulation nodes are connected to world anchors in order to eliminate rigid body modes.

The forces acting on each node in the simulation mesh are given by the gradient of this energy with respect to the nodal degrees of freedom

$$\mathbf{F} = -\frac{\partial E}{\partial \mathbf{x}} \tag{6.2}$$

and the force Jacobian is described by the Hessian of this energy

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = -\frac{\partial^2 E}{\partial^2 \mathbf{x}} \tag{6.3}$$

Similar to the system deformation energy, the forces acting on each node can be divided into forces generated by mesh deformation, contractile elements and pins.

$$\mathbf{F} = \mathbf{F}_{foam} + \mathbf{F}_{contractile} + \mathbf{F}_{pins} \tag{6.4}$$

Foam Simulation Model

The elastic behavior of the foam is modeled using linear finite elements with a compressible Neo-Hookean material model. The deformation gradient of each element e in the simulation mesh is described by $\mathcal{F} = \frac{\partial \mathbf{x}^e}{\partial \mathbf{X}^e} = \mathbf{d}\mathbf{D}^{-1}$. Columns in $\mathbf{d}$ store edge vectors: $\mathbf{d}_i^e = \mathbf{x}_i^e - \mathbf{x}_0^e$, where $\mathbf{x}_j^e$ describes the position of the j-th node of element e in world coordinates. $\mathbf{D}$ defines a matrix containing rest configuration quantities.

For each element, the energy density is then given by

$$\Psi(\mathbf{x},\mathbf{X}) = \frac{\mu}{2}\mathrm{tr}(\mathcal{F}^T\mathcal{F} - I) - \mu \ln J + \frac{\kappa}{2}(\ln J)^2 \tag{6.5}$$

with the identity matrix I, material parameters μ and κ found in Table 6.1, and $J = \det\mathcal{F}$. Integrating Section 6.1 over its domain yields the elastic energy stored by an element, and the elastic energy of foam E_{foam} is computed by summing up element energy contributions of all elements in the simulation mesh.

Table 6.1.: Material properties used in foam simulation model.

ρ $[kg/m^3]$	μ $[Pa]$	λ $[Pa]$
160	1.2e6	2e6

Pins

A surface mesh of a simulated foam model typically consists of 1000 - 2000 nodes, depending on the model geometry. To anchor the simulated foam model in space, *pins* (zero-length springs) are attached to a small number of nodes, defined by the user. In the physical system, these pins correspond to a physical fixation of the foam, e.g. the attachment at the bottom of a foam hand to the base.

Tendons

Tendons are modeled as contractile elements which implement the contraction of a tendon by 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 $(x_1, \dots, x_n)$. All points of contractile elements are assumed to be 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

$$l_0 = \|x_s - x_1\| + \sum_{i=1}^{n-1} \|x_i - x_{i+1}\| + \|x_n - x_t\| \tag{6.6}$$

The contraction level $a_c \in [0,1]$ of each tendon describes the contracted length as $l_c = l_0 \cdot (1 - a_c)$. In the following, the word *routing* refers to the choice of endpoints and intermediate vertices of each tendon.

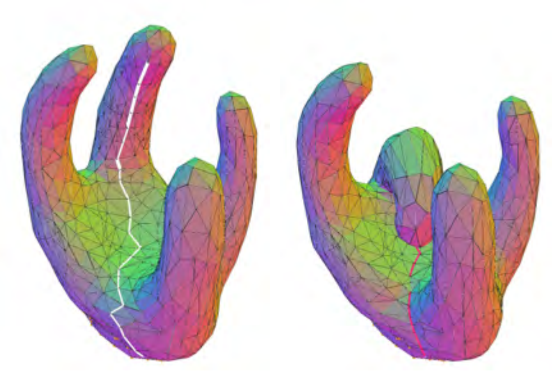


Figure 6.1.: Following the approach of Bern et al. [6], a finite element simulation is used for foam robot hands, where tendon contractions result in contraction of the mesh along the tendon routing. Equilibrium poses before and after contraction are shown. Building on this previous research, simulation parameters are identified and evaluated to match manufactured foam hands (Section 6.3).

Unilateral strain energy of a tendon is modeled as a piece-wise C^2 polynomial, $U(\Gamma)$ as a function of the deformation Γ. This function is zero below a negative threshold, quadratic above a positive threshold, and cubic in between. More details of this modeling approach are found in [5].

The resulting deformation for a tendon routing with the contractions a_c is calculated by minimizing the total energy of the system using a direct sparse LDLT Cholesky solver from the Eigen library. Figure 6.1 shows a four-fingered hand mesh in equilibrium before and after contracting a tendon.

In order to provide the user with an interactive and intuitive simulation tool to explore the capabilities of foam hands, several simulation modules were developed and are described in Section 6.2.

To identify simulation parameters and justify the usage of this simulation model, a motion capture experiment was conducted (Section 6.3), where simulated hand poses are compared to physical hand poses.

6.2. Simulation Modules

To provide the user with an interactive and intuitive simulation tool to explore the capabilities of foam hands, several simulation modules were developed and are described in this section. The framework is programmed in C++, graphical renderings and user interfaces use OpenGL[1] and nanogui[2]

6.2.1. Interactive Creation of Tendon Routings

Once a triangulated mesh[3] is loaded into the simulation, tendon routings can be intuitively created by selecting nodes on the mesh. Startpoint, intermediate points and endpoint nodes are fed to an A* path planner (Appendix A.2), which returns the shortest path along mesh edges on the mesh surface as tendon. After creation, tendon routings consisting of one or several tendons can be saved an reloaded into a simulation sessions.

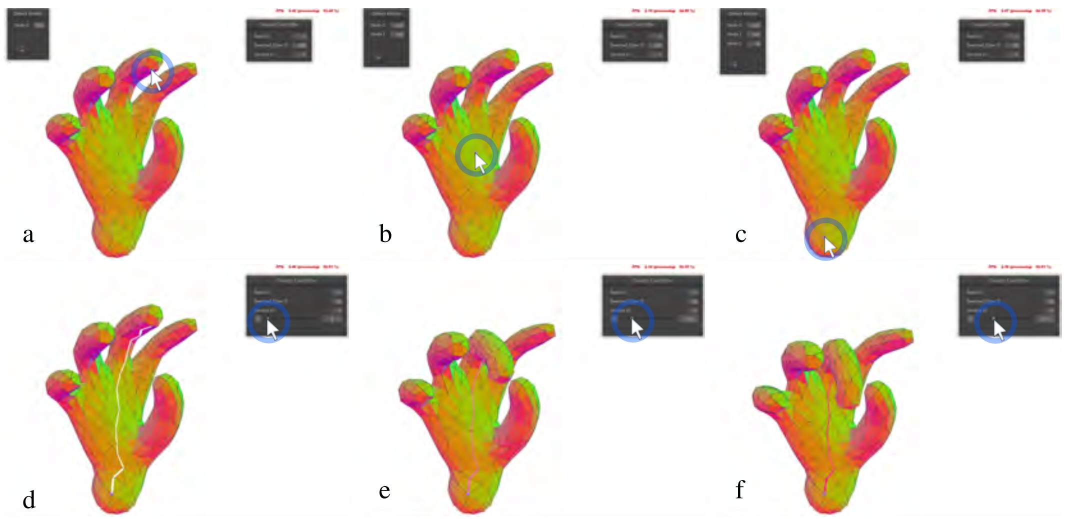


Figure 6.2.: Nodes of the simulation mesh are selected (a-c), and a tendon is automatically routed through the selected points (d). The tendon can be actuated interactively by dragging a slider (e-f).

When a tendon is created, the user is provided with a simple slider to control tendon actuation, ranging from 0% (no contraction) to 100% (tendon fully contracted). At this point it should be noted that contracting a tendon by 100% is not possible for a physical robot as it would imply a tendon length of zero. In simulation however, the contraction level imposes a constraint for the energy minimization, for which the solver attempts to find a stable minimum energy state, resulting in a non-zero tendon length.

Tendons can be contracted using the sliders, and the resulting deformation is rendered within the graphical user interface, shown in Figure 6.2 e-f. While this enables the user to explore the motion capabilities of a hand design, desired tendon configurations and actuations can also be stored as motion keyframes. By looping through interpolated tendon activations, motion sequences are rendered on the

[1] OpenGL, `https://www.opengl.org/`

[2] nanogui, `https://nanogui.readthedocs.io/en/latest/index.html`

[3] TetGen, `http://wias-berlin.de/software/tetgen`

screen. This provides the user with the ability to create, save and reload complex hand posing and motions.

6.2.2. Interaction with Physical Robot Hand

A foam robot hand can be operated by sending tendon contractions from the simulation framework to the corresponding servo motors. The communication between simulation and motors is implemented using the modular ROS software stack described in Section 7.1. Tendons are actuated within the Windows-based simulation and the activation levels are sent as a serialized message to the ROS network via TCP/IP.

6.2.3. Fingertip Control and Pose Generation using a Cyberglove

To deform a hand mesh and create meaningful poses in simulation (e.g. grasps), an intuitive approach is to map a user's hand configuration to the simulated hand mesh. For this purpose, a user poses the hand in the desired configuration, and this configuration is recorded with a CyberGlove[4] featuring 22 resistive bend sensors. Since the simulation model does not contain joints or links, only the 3D position and surface normal of the fingertips is used to map between physical hand an simulation model. For this purpose, on each fingertip that the user wishes to control, one surface triangle (3 nodes) is selected. The centroid of this surface triangle represents the position of a simulated fingertip and the surface normal gives its orientation. Positions and orientations use the coordinate system at the base of the palm as reference, therefore the mapping is invariant to the wrist pose.

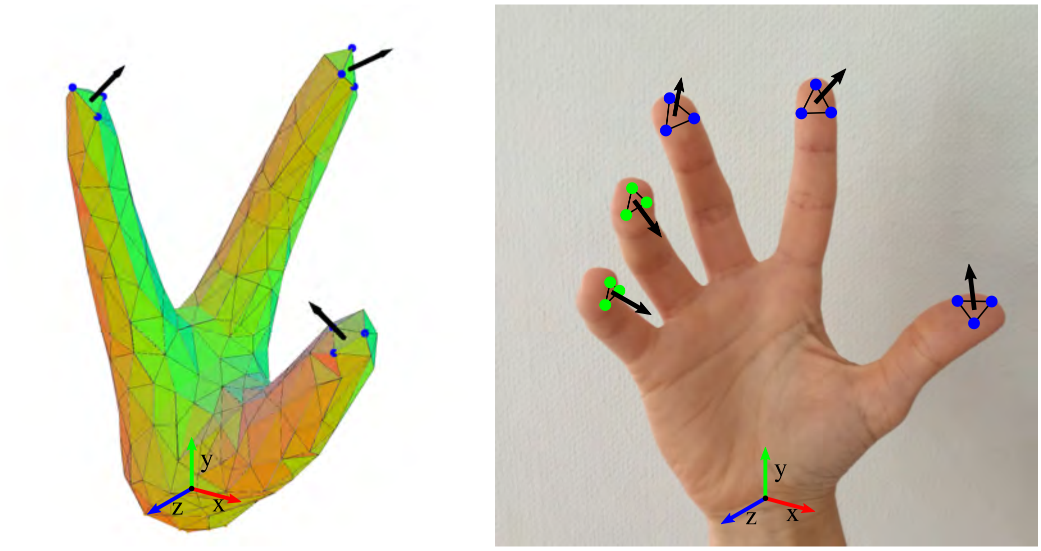


Figure 6.3.: Mapping between positions and surface normals of the human fingertips (right) to triangles on a three-fingered mesh (left). Three surface triangles were selected on the mesh and are controlled using the thumb, index and middle finger. The fingertips of the ring and pinky (green) are not used, but can be included to control a fourth and fifth finger. The coordinate systems of simulated and physical hand are shown at the base of the palm.

The current positions of the operator's fingertips and the corresponding surface normals are extracted using the CyberGlove SDK, and sent via TCP/IP to the simulation framework. In simulation, the user can select the number of fingertips he wishes to control (using one glove, up to 5 fingertips can be mapped), and selects the corresponding triangles on the simulation mesh. The selected triangle nodes are then

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

moved and pinned to the corresponding 3D position in simulation, dragging the finger into the desired configuration.

Figure 6.3 depicts the mapping between the human fingertips (right) to triangles on a three-fingered mesh (left). Three triangles were selected on the mesh and are controlled using the thumb, index and middle finger. This mapping technique can be used to move arbitrarily shaped meshes into a desired configuration.

A posing sequence using this process is shown in Figure 6.4. For this sequence, the fingertips of the thumb, index and middle finger were used to manipulate a three-fingered mesh.

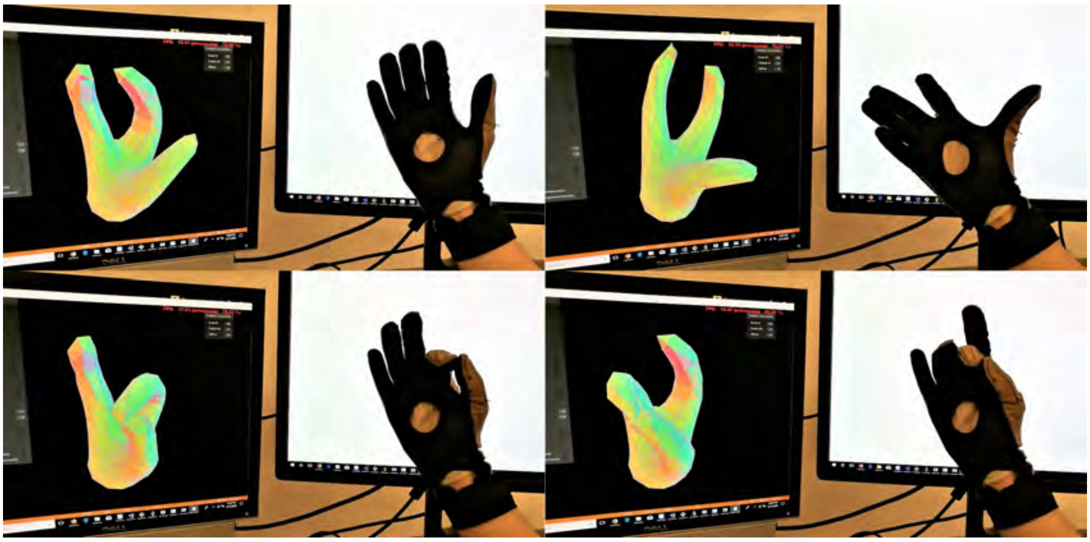


Figure 6.4.: Teleoperated posing of a three-fingered simulation mesh using a CyberGlove. The fingertips of the thumb, index and middle finger are mapped to selected surface triangles on the mesh.

6.3. Validation using Vicon Motion Capture

In this section the accuracy of the simulation framework is quantified by comparing fingertip trajectories of a simulated and a physical foam robot hand. The deformations of the foam are tracked using a Vicon Motion Capture system. Predicted deformations from simulation, are then compared with the corresponding actual deformations on the physical robot.

Four-Fingered Foam Hand

The physical foam robot hand used in this experiment is a non-anthropomorphic hand with four fingers and 10 tendons. Each finger is controlled by a pair of antagonistically routed tendons acting as flexor and extensor. In order to introduce abduction and adduction motions, two additional tendons were placed on the left and right side of one finger. As reference in simulation, the same geometry was used, with a slightly coarser mesh (980 nodes) compared to the mesh that was used to print the mold of the physical foam hand. This mesh size was chosen to allow interactive simulation in our user interface.

Motion Capture Experiment

Fingertip trajectories of the four-fingered foam hand are recorded using a Vicon motion capture system with 12 cameras. To get a robust estimate of the position and to prevent occlusions, four markers are placed around each fingertip as shown in Figure 6.5. For registration purposes we additionally place

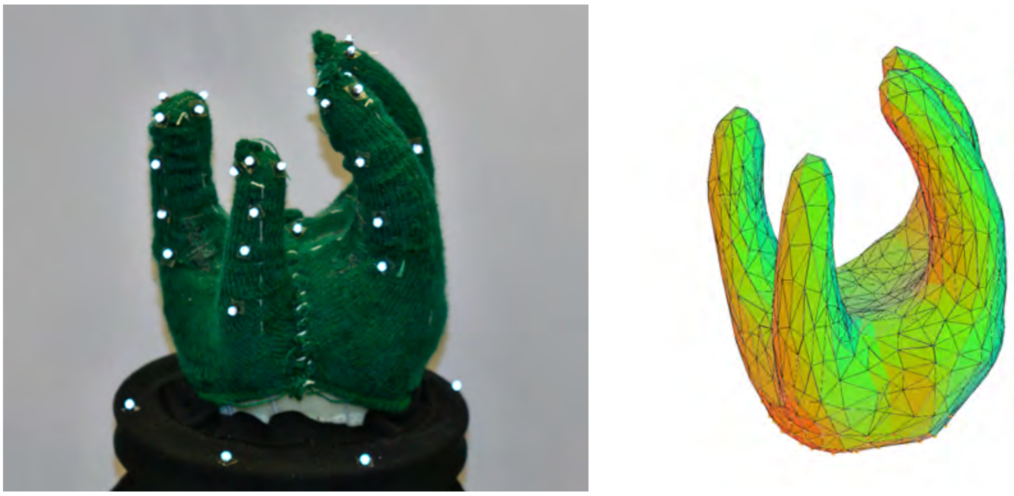


Figure 6.5.: Left) Four finger hand with markers. Right) Simulation mesh of the same hand model. The simulation mesh is anchored in the world coordinate system by pins on the bottom nodes of the mesh.

markers on the platform on which the hand is mounted and alongside each finger. After the experiment the recorded markers are registered on the 3D mesh. This is done using a standard ICP algorithm[5] that minimizes the distance of points from the mesh and the markers with respect to each other. Since the markers themselves are not exactly aligned with the surface of the foam, it is difficult to infer the exact position of the fingertip using only the position of the markers. Therefore each fingertip position $\vec{p}_j$ with $j = \{1,\ldots,4\}$ is defined as the mean of the corresponding markers k with $k = \{1,\ldots,4\}$, with a distal offset of 5mm normal to the plane spanned by the four markers:

$$\vec{p}_j = \frac{1}{4} \cdot \sum_{k=1}^{4} \vec{p}_{jk} + 0.005 \cdot \vec{n}_j \tag{6.7}$$

The RMS error describing the euclidean distance between the aligned point clouds of our ICP registration was 4.05mm.

Table 6.2.: Material properties used in FEM Simulation

ρ $[kg/m^3]$	E $[Pa]$	ν
160	3e6	0.25

In terms of material parameters for the FEM simulation (mass density ρ, Young's modulus E, Poisson's ratio ν) the applied values can be found in Table 6.2.

The goal of this experiment is to give an estimate of how well the simulation can match reality. 5 trials were run in which each tendon is repeatedly contracted from 0% to 50% of its rest length in steps of 10%. The tendon rest length is distinct for each tendon and is computed in simulation.

[5] MATLAB, MATLAB Toolbox For C3Dserver `https://www.c3d.org/appmatlab.html`

Results and Discussion

A motion sequence of a contracting extensor tendon moving the simulated hand through the waypoints at 10%, 20%, 30%, 40%, and 50% contraction is shown in Figure 6.6. The motion is displayed from three different camera views. The fingertip trajectories recorded by the Vicon system are marked as dotted lines, with larger green points at the fingertip positions recorded at the tendon contraction waypoints (10%, 20%, 30%, 40%, 50%). In each frame, fingertips of the simulated hand are marked as red circles. From this sequence it can be observed that trajectories of the simulated and physical fingertips largely coincide.

The resulting error between fingertip positions captured with the Vicon system and from simulation is depicted in Figure 6.7. The mean position error for all fingers including all activation levels is 0.626cm. For each individual finger median error and the quartile deviations are similar at all contraction levels. This suggests that even large deformations do not significantly decrease the accuracy of the simulation.

In general we identify the following sources of position errors:

- small deviations between tendon routings in simulation and reality
- tendon slack
- registration errors in motion capture system
- friction between tendon and glove
- slight relative movements of foam core and glove during actuation.

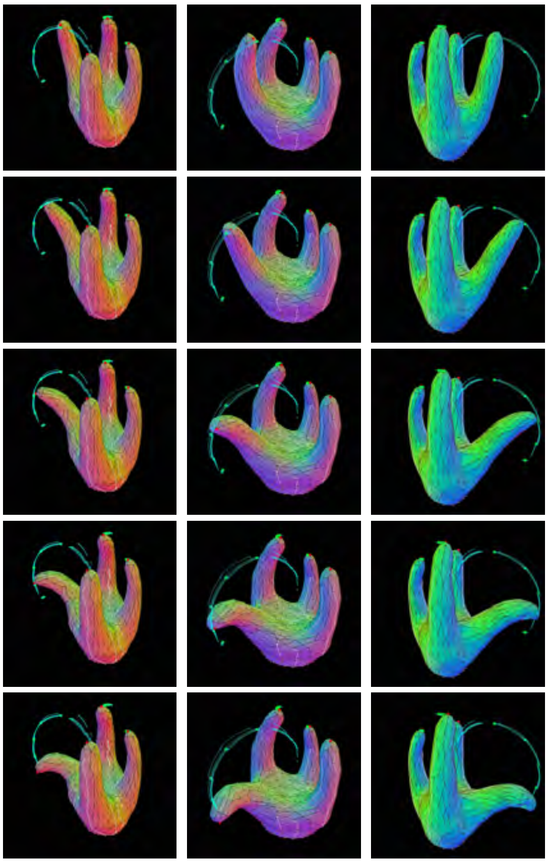


Figure 6.6.: Simulated hand and motion capture trajectories for an extensor tendon moving from 10%(top) to 50%(bottom) tendon contraction, viewed from three different camera perspectives. Fingertip positions recorded by the Vicon system at each contraction level are averaged over all five trials and marked as green dots, fingertip positions of the simulation model are marked as red dots.

The significant difference in position error between finger 1 and the remaining fingers (Figure 6.7) suggests that the physical routing of at least one tendon on finger 1 differed significantly from the simulated routing.

Most of the described errors can be mitigated during fabrication of the hand, for example by using teflon-coated tendons or different gluing techniques. The accuracy of the manual sewing procedure to realize the tendon routings on the physical robot could be enhanced by embedding 3D-knitted tendon waypoints in the glove using automated knitting processes [74]. A further approach could embed routing points in the form of flexible hooks directly in the foam, e.g. by designing cavities in the 3D printed mold, and placing hooks in these cavities before casting the foam.

The results of this evaluation suggest that the model predicts foam deformations sufficiently well and can be applied to explore tendon routings and create desired target poses (Section 6.4). The results additionally encourage the application of simulation based learning of mapping from desired pose to tendon actuation (Section 7.5).

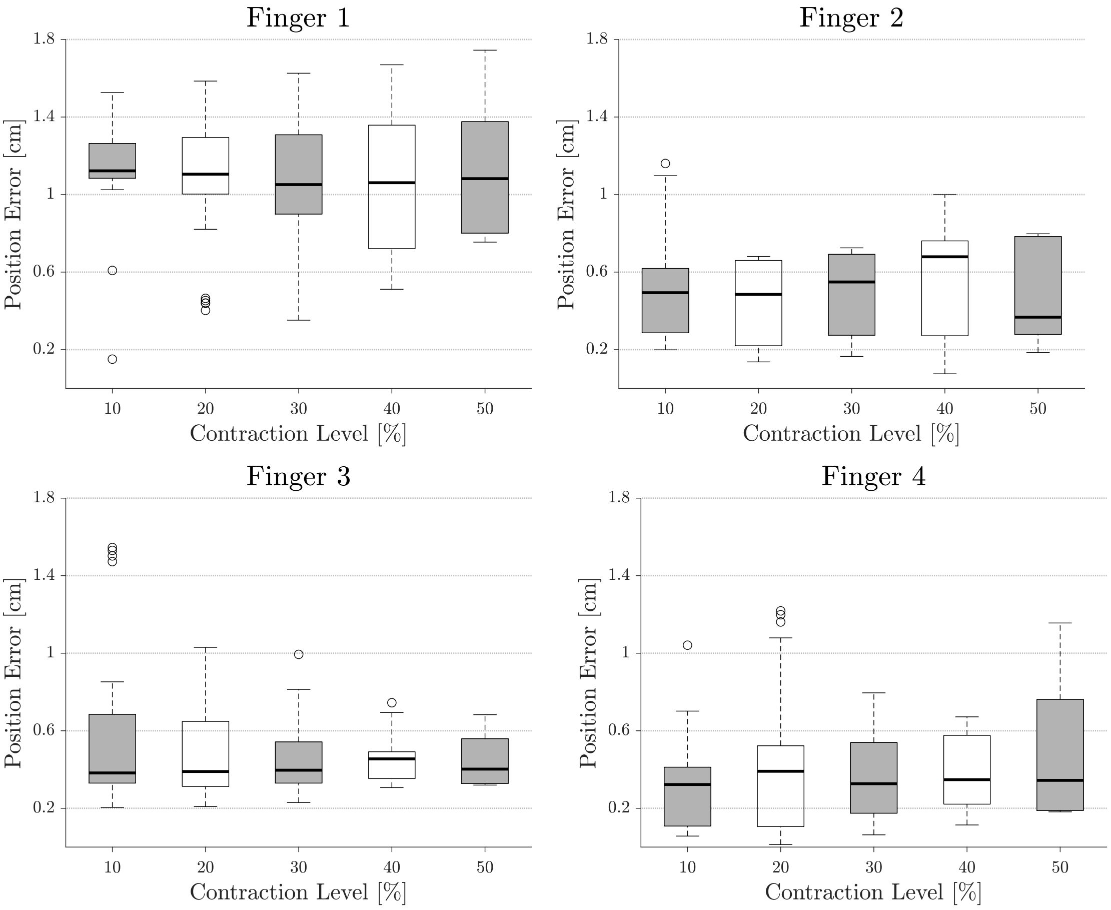


Figure 6.7.: Position Error between simulated and captured fingertip positions of finger 1-4 at respective contraction levels.

6.4. Application: Human Subject Study on Tendon Routing Design Ability

Using the developed simulation framework, a user study was conducted to determine the quality of manually designed tendon routings for foam hands. The ability to easily create foam hands and tendon routings that can achieve desired poses is an important aspect to make soft foam hands accessible to everyone, which is a major goal of this work. Additionally, this study serves as a benchmark for the automated design of foam hands [3]. In this work, the study results are compared to automatically generated tendon routings in order to evaluate the quality of the automated design process. The following sections describe the questions and purpose of the study and the experiments that were conducted. Furthermore, the results are presented and discussed.

6.4.1. Questions

The goal of this study is to quantitatively measure human ability to manually create tendon routings for a desired hand pose of a foam hand.

The central questions explored in this study are:

- How good is the quality of human designed tendon routings?

- Can anyone design tendon routings?
- How do manual tendon routings compare to automatically generated designs?

6.4.2. Experiments

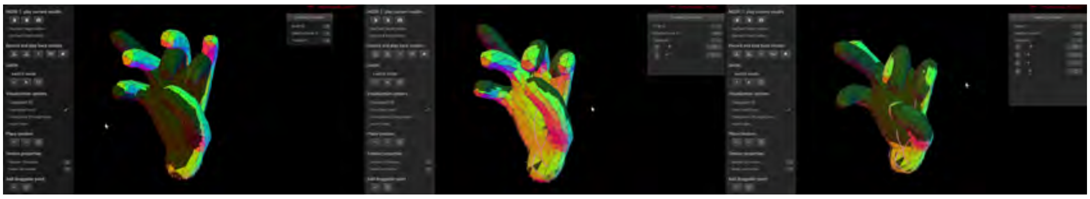


Figure 6.8.: Screencast of a user creating an exemplary tendon routing. The user's task is to place and contract tendons to match the target hand pose (depicted in darker colors). The current pose of the simulation mesh is shown in bright colors. Within the sequence, the user places four tendons on the mesh and select contraction levels using the slider toolbar on the top right.

10 participants were asked to create tendon routings and corresponding actuations to move a simulated hand into desired configurations. The test subjects were asked to confirm that they are not and have not been working/researching in the field of soft robotic hands before participating in the study. For the simulated hand mesh, a cupped rest pose shown in Figure 6.9 was chosen to serve as initial mesh. This mesh was created using the automatic hand mesh creation process described in Bauer [3].

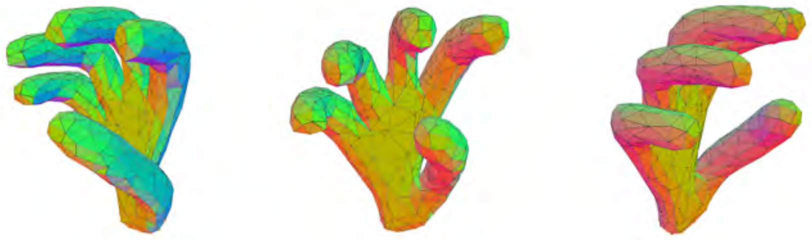


Figure 6.9.: Cupped rest pose of a five fingered simulation mesh. This rest pose was used in the user study.

14 Target grasps from the cumulative taxonomy (Section 2.2.4) were recorded using a CyberGlove and the mapping process which is described in Section 6.2.3. The recorded grasps are transferred to the simulation poses shown in Figure 6.10. For each grasp, the left image shows the target grasp with a CyberGlove, and on the right, the configuration of the deformed simulation mesh is shown. The configuration of each finger is characterized by the 3D position of the fingertip and the surface normal of the fingertip, depicted as a black arrow. Positions and normals for each pose are computed using the virtual human hand model provided by the CyberGlove SDK. From the 14 recorded grasps, 3 grasps were selected as target poses for the study (the number in parentheses refers to the number of the grasp in Figure 6.10):

1. Lateral Tripod (3.2)
2. Medium Wrap a (1.1a)
3. Prismatic 3-finger (6.3)

The participants were asked to create a tendon routing within 10 minutes per pose, and they could use up to 10 tendons per pose. The graphical user interface is shown in Figure 6.8, where a user is recreating an exemplary target pose by adding and contracting tendons. The target pose is depicted in gray, and the current pose of the hand mesh is shown in bright colors.

6.4.3. Results and Discussion

To quantify the quality of a tendon routing, for each finger the euclidean distance between desired and current fingertip is computed. This position is given by the position of the centroid of the selected surface triangle on each fingertip. Additionally, the fingertip surface normals of the target pose and the current pose are compared by calculating the angle between the two normals. A complete list of the study results (images of poses, distance error and angle between normals) can be found in Appendix A.3.

Over all poses and for all test subjects, the average distance error between the created and the target fingertip was 1.378 cm, and the average angle between normals was 64.6 degrees.

The high deviation between surface normals of fingertips can be attributed to the computation method: Since only the surface normal of one triangle is considered, the result can be heavily influenced if a tendon is anchored close to the triangle. If a tendon is routed nearby this triangle, a contraction of the tendon will significantly change the orientation of local triangles. This could be mitigated by instead using the average orientation of surface normals of several triangles on the fingertip of the mesh. A further approach would be to compare contact points and contact forces of the hand mesh with a (simulated) grasped object. However, this requires the implementation of a contact simulation to model the interaction between soft foam hands and rigid or deformable objects. Such a simulation is not available at this time.

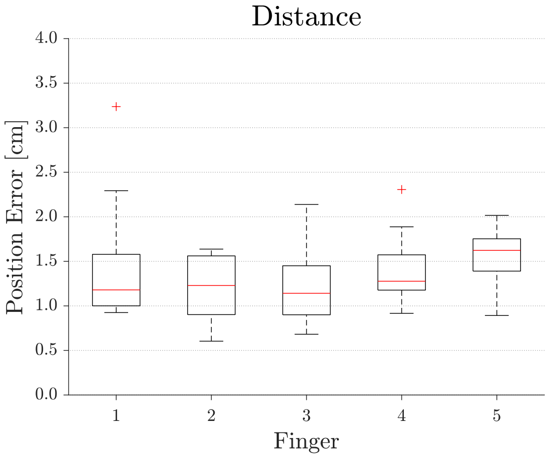


Figure 6.11.: Mean position error of tendon routings designed by human test subjects. For each target pose, the distance between the fingertips of the target pose and the pose created by each test subject is computed. For each test subject, the mean error of each fingertip is computed over all 3 target poses.

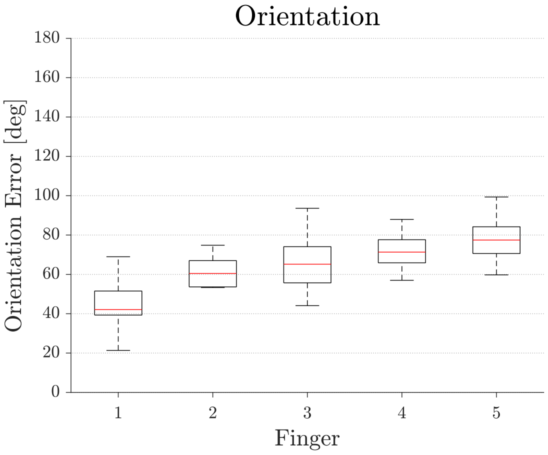


Figure 6.12.: Mean orientation error between target and achieved surface normals on fingertips. For each target pose, the angle between the surface normal of fingertips of the target pose and the pose created by each test subject is calculated. For each test subject, the mean angle of each fingertip is computed over all 3 target poses.

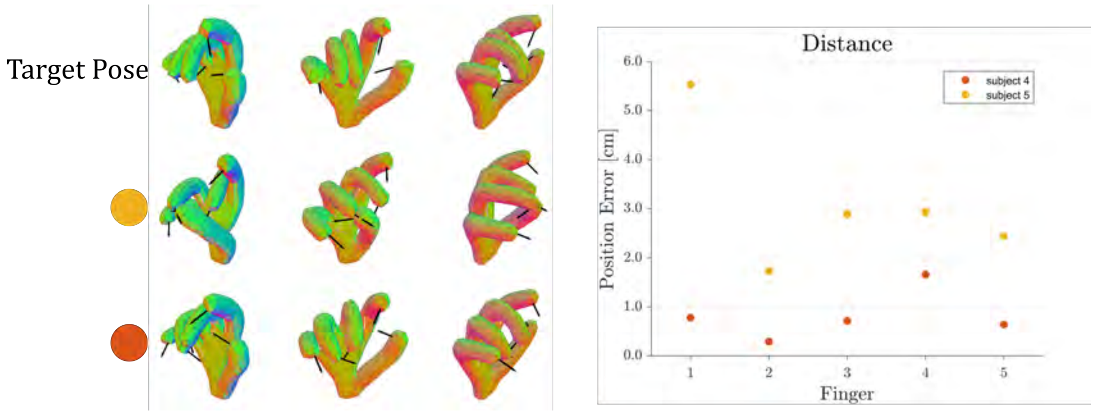


Figure 6.13.: Target pose and two examples of poses created by test subjects for target pose 1 (Lateral Tripod).

The distribution of position errors for the five fingers of the hand (1 - thumb, 2 - index, 3 - middle, 4 - ring, 5 - pinky) are depicted in Figure 6.11, the distribution of orientation error is shown in Figure 6.12.

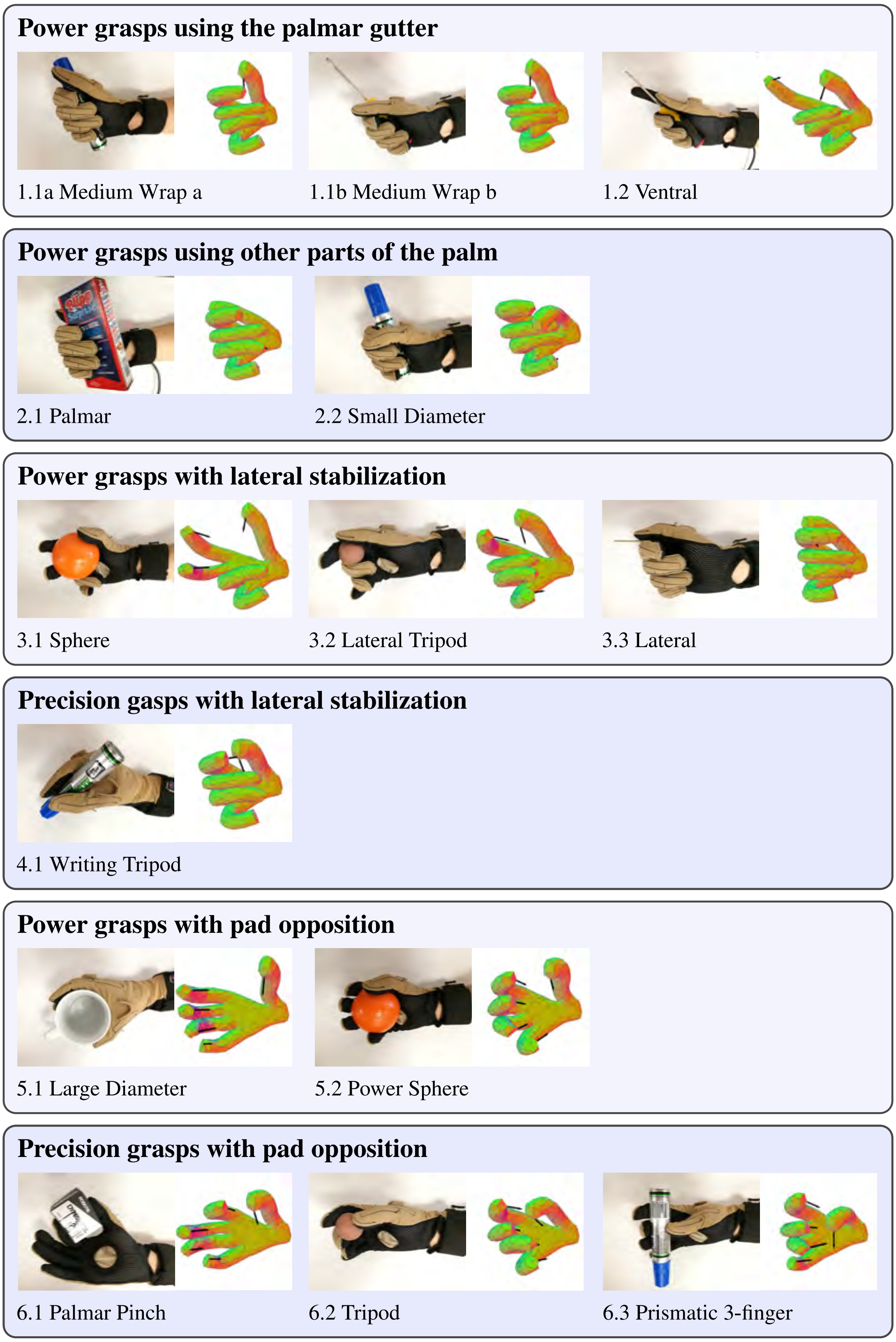
Taking into account the finger lengths of the simulated hand (10-12cm), the achieved mean position error of 1.379cm accounts for 11% - 14% of finger length. It is therefore relatively low, and from the images in Appendix A.3 it can be observed that the poses created by the study participants qualitatively match the target hand poses closely. These results suggest that with the provided simulation tools, users can intuitively and quickly create tendon routings and actuation levels that lead to a qualitatively good match with a desired target pose. It can further be noted that some test subjects seem to have a better intuition than others about the mesh deformations that they are creating when placing and contracting a tendon. Figure 6.13 shows two examples of poses created by test subjects for target pose 1 (Lateral Tripod). While the design by test subject 4 (highlighted in red) matches the target pose very closely, the pose created by test subject 5 (highlighted in yellow) deviates noticeably from the target. Especially in thumb and middle finger configurations this divergence is evident.

6.4.4. Limitations and Outlook

In order to further evaluate the study results regarding success/failure of grasps, the designed routings have to be transferred to a physical foam hand system. The following characteristics are identified as requirements for this transfer:

- Tendon Routing must be realizable: As an example, a simulated tendon can end at the middle of the palm, while in the physical system, tendons have to be connected to motors which are housed away from the hand. The tendon has to be routed to the bottom of the hand, causing different and additional deformation compared to the simulated routing.

- Grasp must not fail: Although a pose is matched in terms of fingertip position and orientation, the grasp might still fail. Matching the fingertips might not be a sufficient measure to create a successful grasp, especially if the grasp involves contact points or areas at different areas of the hand (e.g. the palm). Solving for matching contact points and forces may be a solution to this problem. However, this requires a realistic contact simulation to model the interactions between soft foam hands and objects.



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 6.10.: 14 Target grasps recorded with a CyberGlove, transferred to simulation poses. For each grasp, the left image shows the target grasp, and on the right, the configuration of the deformed simulation mesh is shown, including surface normals (black arrows) of fingertips. *Image by Dominik Bauer.*

7. Operation of Tendon-Driven Soft Foam Robot Hands

In this chapter, drivers and control mechanisms are presented which are applied to operate foam robot hands. First, the modular ROS control software is described. This software is then used to conduct a series of experiments to explore the capabilities of foam hands, such as repeatability, strength and grasping and in-hand manipulation capabilities. An intuitive approach to teleoperate the robot using a CyberGlove is applied to an anthropomorphic hand. In the final section, learning-based methods to solve the inverse kinematics problem are tested and compared in simulation.

7.1. Modular ROS Control Framework

To operate foam robot hands, a modular control framework based on ROS was developed. The main goal of this software is to provide a practical and modular structure to enable rapid testing of new modules and switching between operation modes (e.g. create keyframe poses for grasp sequences, play back demonstration showcases, teleoperation via a CyberGlove). Figure 7.1 shows the relations and interactions between the software components.
The following modules were created:

- **Motor Communication:** This node implements a position controller for multiple motors based on the Dynamixel SDK. A subscriber is set up to track messages that are published to the '/mapping/desired_state' topic. If a message is received from this topic, the command is sent to the motors.

  The motor communication node additionally publishes the current state of the motors to '/motor_com/dynamixel_state'. Alternatively, for debugging purposes, a full diagnostic message is sent to '/motor_com/dynamixel_full_state'.

- **Mapping:** Tendon actuations α_{ai} with $i \in \{1,\ldots,n_{tendons}\}$, coded as contraction lengths in m, are converted to motor positions $\alpha_{mi} \in [min\_position,\ldots,max\_position]$. This conversion depends on pulley diameter and position unit of the motors. For Dynamixel AX-12A motors, the motor can be moved within a range of 300 degrees, and the corresponding position values range from 0 to 1023, hence the unit is 0.29°.
  - **Teleoperation via CyberGlove:** In this mode, the mapping node subscribes to the topic '/sensor_msgs/cyberglove'. A pre-trained regression model (Section 7.4) takes joint angles measured by the CyberGlove as an input and predicts the corresponding motor positions, which are then published to '/mapping/desired_state'.
  - **Teleoperation via Simulation:** The mapping node receives tendon actuations on the topic '/ext/python_socket_out', converts them into motor positions and publishes motor positions to '/mapping/desired_state'.

- **Read CyberGlove:** The CyberGlove is connected through a serial port and the joint angle values measured by the glove are published to '/sensor_msgs/cyberglove'.

- **Python Server Socket:** Since the FEM simulation framework (Chapter 6) is built in Microsoft Visual Studio and runs on Windows, no straightforward approach to communicate via ROS is available. Instead, the communication between simulation framework and the ROS network is implemented via sockets (TCP/IP): The user actuates tendons in simulation to achieve the desired pose. The activation levels are then sent from the client socket (Windows socket, C++) to the server socket (linux python socket), which is part of the Python Server Socket ROS node. This node then publishes the received actuations to '/ext/python_socket_out'.

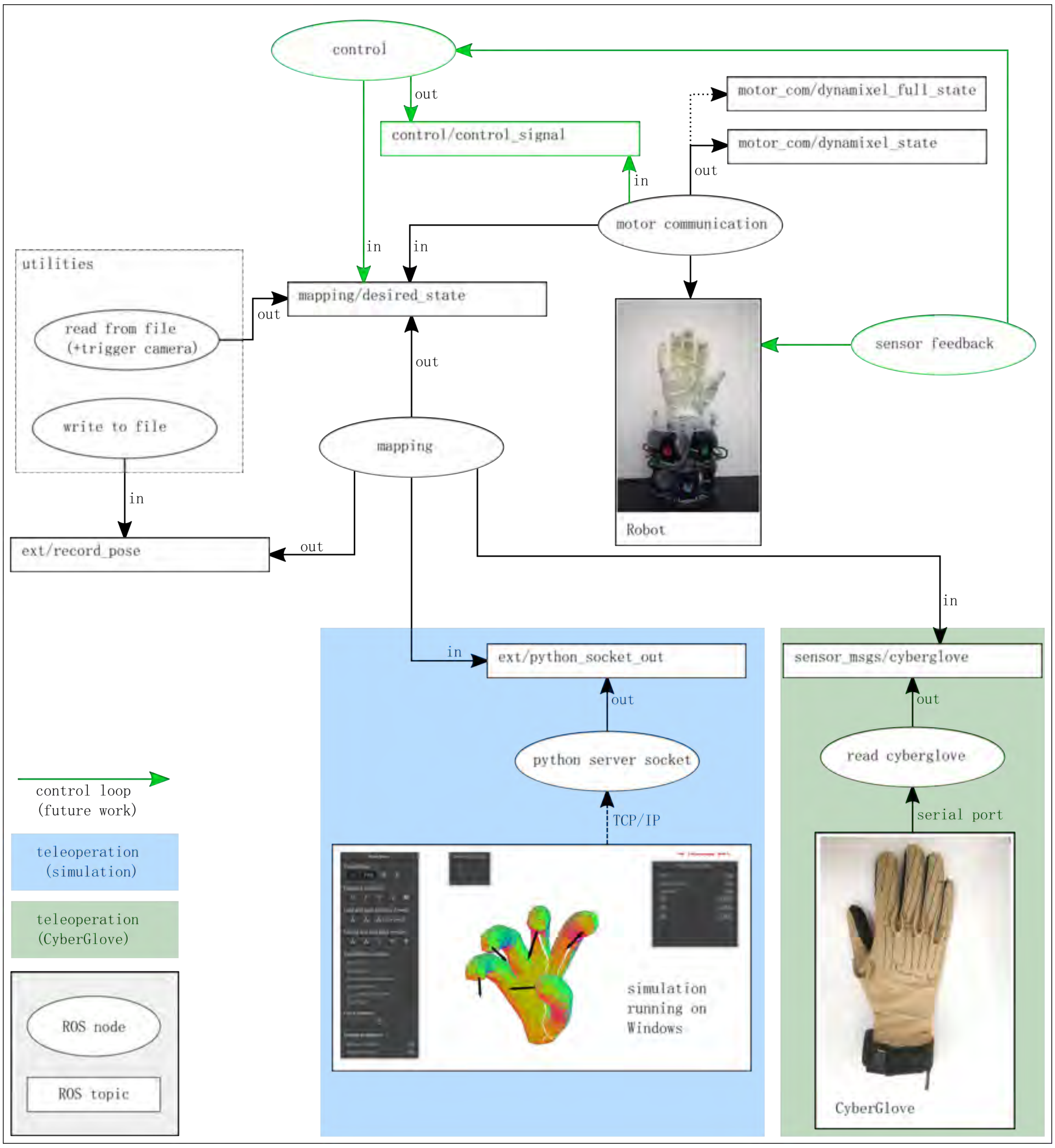


Figure 7.1.: Modular ROS robot control framework. Nodes publish messages to topics ('out') and subscribe to topics ('in') to receive messages.

- **Utilities:** This package includes functions to record poses from the physical robot and write the corresponding motor actuations to a file. Saved keyframe poses can be played back to create motion sequences.

  Defined keyframe poses, provided by the user as a .txt file, can be loaded and sent to the motor controller. This enables the user to create desired poses, interpolate between them and have the robot loop through the poses. Additionally, a timed camera capture can be triggered synchronously to the motor commands, to record images of the robot in its configured poses.

- **Sensor Feedback:** This module enables the integration of sensor feedback.

  As an example, 10 flexible resistive bend sensors[1]were placed in pockets which were custom sewn[2]on an off-the-shelf glove fitted on the foam hand (Figure 7.2). Analog sensor values were recorded using an Arduino Mega 2560 and the Arduino IDE. This approach was not further investigated due to very low sensor readings resulting from non-optimal sensors placement on the hand: The sensor pockets were placed at locations on the foam hand which correspond to human hand joints. The elastic foam hand however exhibits deformations fundamentally different from the human hand since deformations are not strictly limited to joints, but occur over large areas. This causes relatively small bending angles of the sensors and hence very low sensor readings.

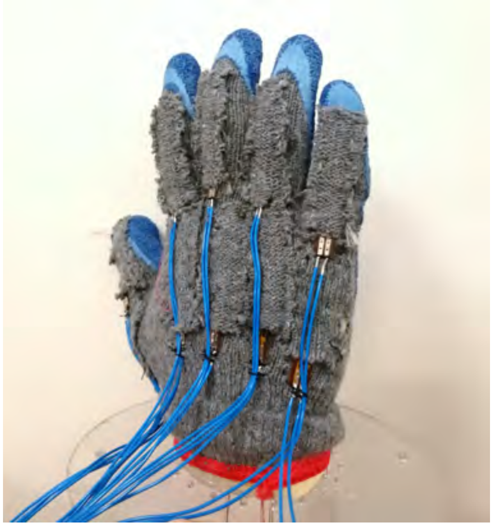


Figure 7.2.: Custom sensor glove on a foam robot hand.

- **Control:** This module has not yet been implemented, but can be used in future works. It is intended to provide the infrastructure to compare the current pose (e.g. using optical sensors, or using more/differently placed bend sensors attached to the glove) with the target configuration. Different control schemes can be tested and implemented in future projects.

7.2. Experiments: Repeatability, Strength, Grasping and In-hand Manipulation

Repeatability

Using the ROS control pipeline described Section 7.1, a repeatability experiment was conducted with a planar gripper robot. This is joint work with Jonathan King and Yuzuko Nakamura, who were responsible for gripper fabrication and hardware assembly, the creation of grasp poses and data analysis.

To investigate repeatability of foam robots, a planar gripper was fitted with 2 flexor and 2 extensor tendons. 7 black target points with a diameter of 6mm were adhered to the textile skin of the gripper to track motions. 6 different grasps were created by moving the robot into the desired configuration and extracting the associated tendon actuation levels. The corresponding tendon actuation levels were then repeatedly applied, and a synchronous image capture was triggered to obtain images of the gripper using a webcam[3]. Images of 800 trials of each grasp were collected and analyzed[4].

Assuming a ‘break-in’ period, the first 50 trials were discarded. 50 random trials for each of the 6 grasp were selected for analysis from the remaining 750 trials. This was due to limited video processing power and time constraints. The markers on the textile skin were tracked by applying a Grayscale Conversion, Gaussian Blur, Prewitt Edge Filter, and Hough Circle Transform, in sequence to each frame using MATLAB. The circles were then sorted using *Nearest Neighbors*.

[1] https://www.robotshop.com/en/2-unidirectional-flexible-bend-sensor.html

[2] Textile patches were sewn onto the glove by Kai-Hung Chang

[3] Logitech 1080p Webcam

[4] Repeatability analysis and figures by Jonathan King

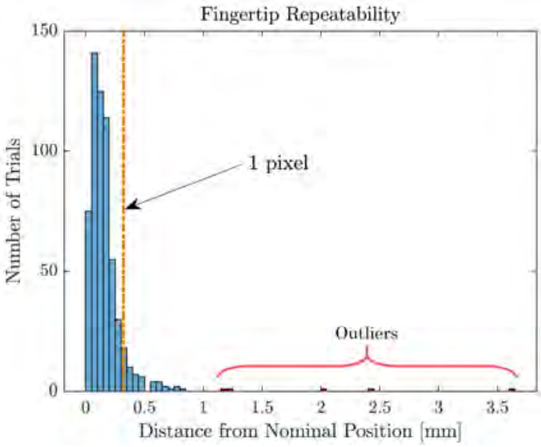


Figure 7.3.: Histogram of fingertip repeatability over 600 trials.[4]

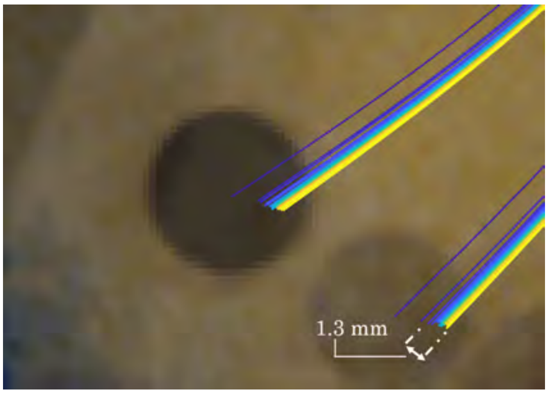


Figure 7.4.: Drift of the fingertip position over 800 trials, from blue to yellow.[4]

Figure 7.5 shows three of the grasp poses superimposed for all 50 trials and at 6 different steps along the trajectory. Splines were fitted to the markers for each image.

Repeatability analysis was conducted considering only the fingertip markers at the final stage of each grasp, since they are subject to the most significant displacement. For each grasp $g \in \{1,\ldots,6\}$ and each trial $t \in \{1,\ldots,50\}$ the error was computed as the deviation $e_{gt} = \|p_{gt} - \overline{p_g}\|_2$ from the mean $\overline{p_g} = \frac{1}{50}\sum_{i\in t} p_{gt}$ over the trials. A histogram of the errors for all 600 data points (300 trials, 2 fingertips) is shown in Figure 7.3. Several outliers in the data are suspected to be caused by rare instances of faulty serial communication via serial port. The servo motors do not receive a motor command and therefore remain in their previous position. For future experiments, a solution to this problem is to include a simple switch to check if the motor command was received and executed, and if it was not, resend the command. Table 7.1 shows the distribution metrics considering all trials and considering only inlier data, computed with a conservative $\mu \pm 3\sigma$ filter. Since many measurements were sub-pixel in length, future experiments should use a higher resolution camera or high resolution motion capture.

The fingertip positions are plotted in order in Figure 7.4, revealing a drift over time across the 800 trials. A possible reason for this drift is yield (stretch) in the textile components, which could be reduced by the usage of stronger yarn in the knitted gloves of future designs.

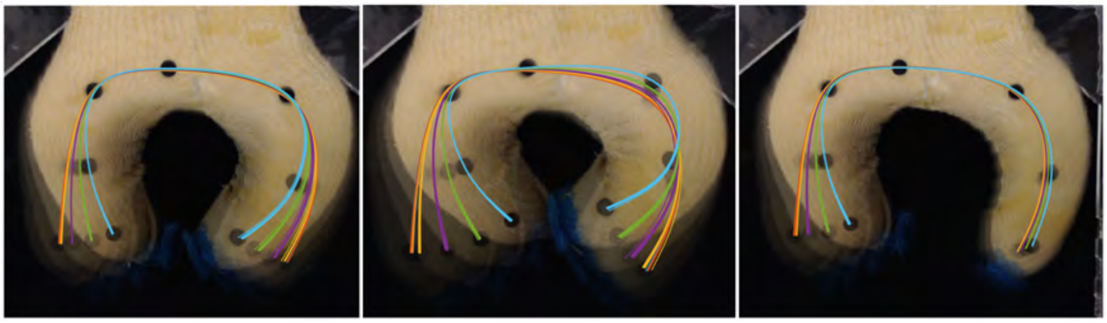


Figure 7.5.: Three different grasps, 6 steps along the trajectory superimposed for each grasp. Splines are fitted to the markers for each image of 50 trials.[4]

Table 7.1.: Repeatability statistics for planar gripper.[4]

Values in [mm]	μ	σ	**median**	**max**
All Trials	0.1738	0.2293	0.1307	3.6360
Inlier Trials	0.1576	0.1210	0.1296	0.8160

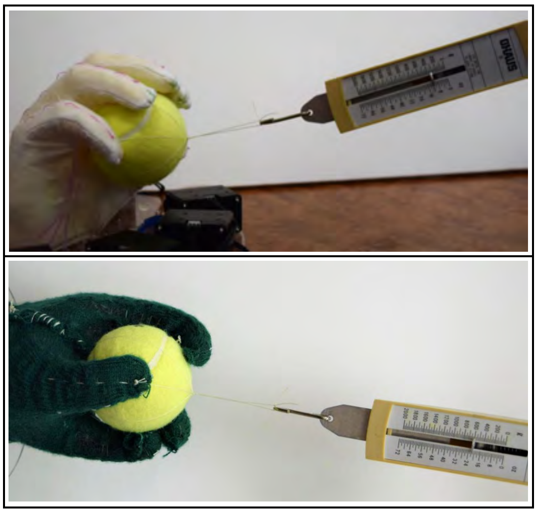


Figure 7.6.: The strength of caging grasps was measured by pulling on a grasped tennis ball until failure. Top: Anthropomorphic hand, 3.2N. Bottom: Four-fingered gripper, 5.8N.

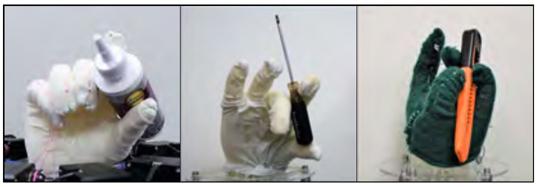


Figure 7.7.: Demonstration of static grasping with a glue bottle (left), a screwdriver (middle) and a box cutter (right).

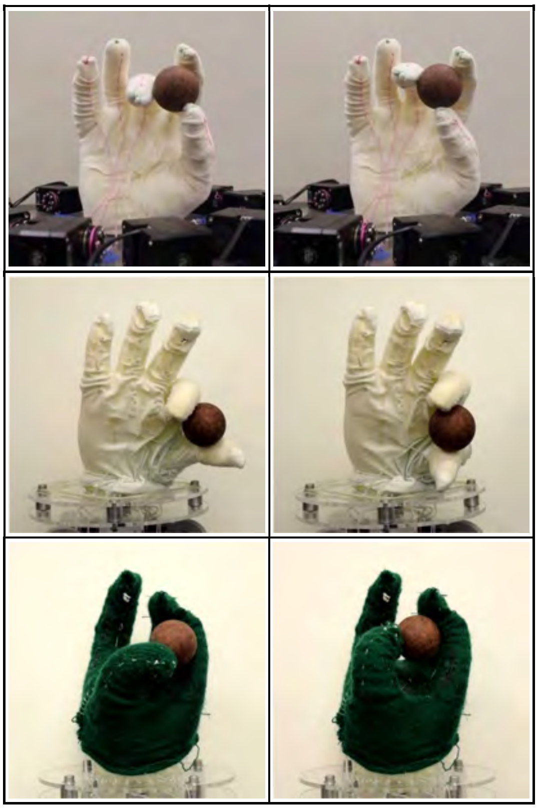


Figure 7.8.: Three distinct robot hands performing precision in-hand manipulation (twisting a ball). Smooth motions are created by interpolating between motor actuation keyframes.

Strength

Grasp strength was evaluated for caging grasps of a tennis ball with two different hand designs: The first prototype (an anthropomorphic hand design) and a four fingered gripper. The test setup is shown in Figure 7.6. The measured pullout force was $3.2N$ for the anthropomorphic hand (top), and $5.8N$ for the four fingered hand (bottom). Since the main difference between the two designs is their geometry, this suggests that a more opposable thumb is important to achieve stronger power grasps with foam hands.

Several of the grippers presented in this work have been in use for over one year and thousands of trials. They have additionally been transported transcontinentally in checked luggage and exposed to harsh weather conditions, all without a noticeable lack in performance. The grippers demonstrate therefore good longevity and ruggedness overall.

Grasping and Manipulation

The developed prototypes are able to perform a variety of grasps and in-hand manipulations. Exemplary static grasps are depicted in Figure 7.7, they are created by playing back a previously created tendon contraction configuration. Several in-hand manipulations are shown in Figure 7.8 and Figure 7.10 on the right. These motions are created by linear interpolation of motor actuations between keyframes, generating smooth transitions between grasps.

7.3. Weaknesses and Current Limitations

The low stiffness of the foam limits the grasping force that can be applied by a foam hand. Especially for ’pushing’ and ’pressing’ tasks, or when a heavier object is to be grasped, this problem is apparent. Future work could address this issue by embedding structures in the foam core during the casting process. A rigid skeleton could increase grasp strength significantly, but defies the principle of an entirely soft hand. A further approach is to embed compartments filled with granular material and leverage the granular jamming technique to achieve variable stiffness. A downside to this method is the need for pressure generating equipment, valves and tubing.

A further limitation to the current foam hand design is the tendon routing method. This method effects the performance of the hands in two aspects. 1) Geometrically: Tendons are restricted to run on the surface of the hand, limiting the poses and motions that can be achieved. If instead they were allowed to be routed through the foam body, the achievable workspace of the hand could be extended and higher forces could be applied by the hands. Past attempts to route tendons through the foam led to tearing of the foam core. This could be avoided by improved manufacturing techniques in future work, for example by embedding tubing into the cast foam. 2) Mechanically: By pulling on the glove, the tendons strain the adhesive between glove and foam core, limiting the force that can be applied to a tendon without failure. If no adhesive is used, gloves tend to slip on the foam core. Future work could explore different materials such as high performance textiles and adhesives to increase performance and durability of foam hands.

7.4. Teleoperation

In the most basic scenario, only the robot itself is available, with a given arrangement of tendons and motors, and a device with which the user wishes to control the robot. With this equipment, a mapping must be learned from user gestures or poses to motor actuations that deform the robot in the desired manner. A straightforward mapping is explored, where the user wears a CyberGlove and controls an anthropomorphic hand that is similar to their own. However, flexibility is desired, when the geometries of the human and robot hands may differ significantly.

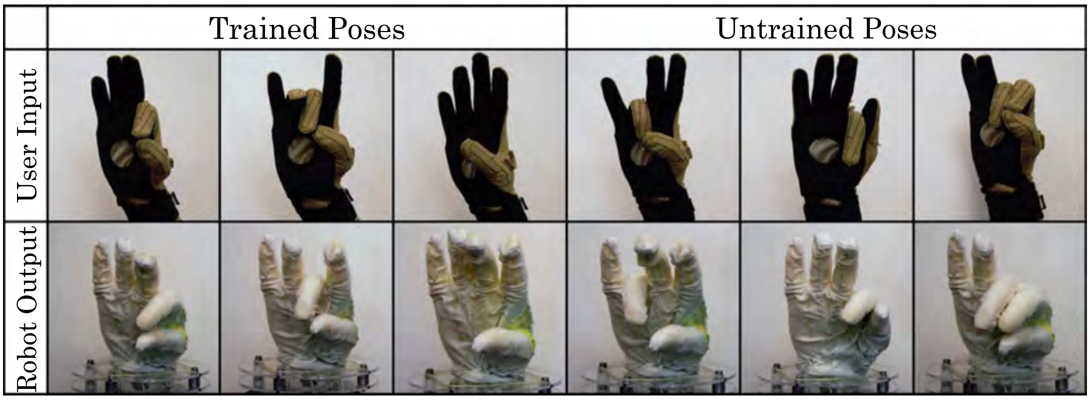


Figure 7.9.: Top) Input poses from user wearing a CyberGlove. Bottom) Output poses from the learned mapping. Left) Poses taken from the training set. Right) Poses not included in the training set.

For the mapping process, inspiration can be drawn from research on puppeteering in computer graphics. For example, Seol and colleagues [101] present a method that allows the user to specify how they wish to move in order to create certain character motions. As an example, they might choose to swing their arm to move an elephant's trunk. In the case of [101], an approach based on feature mapping is used to convert from user motion to character control parameters. For foam robot hands, linear regression is used to create a map from CyberGlove sensors to tendon activations for the hand.

This approach works as follows. First, a sampling of tendon activations is used to execute various poses of the foam hand. An operator imitates those poses while wearing the calibrated CyberGlove, and the corresponding joint angles of the human hand pose are recorded. Both random tendon activations and tendon activations corresponding to finger-thumb oppositions and grasping postures were used to build this training set. For generalization purposes each pose was recorded 5 times.

A regression model, which takes the 22 joint angles from the CyberGlove as input and predicts the corresponding tendon activation levels was trained. The model uses Kernel Ridge Regression with a linear kernel. The average RMS error achieved by the model between the measured and the predicted normalized tendon actuations was 0.0026, with normalized tendon actuation ranging from zero to one.

Even with a small training set (120 recordings), the learned model was able to reproduce a variety of poses with high accuracy.Figure 7.9 shows a comparison of poses supplied by an operator and the poses realized by the foam hand. Both poses taken from the training set and new poses are included. One observation from the proposed teleoperation method is that in order to achieve such results, the careful selection of training poses is crucial. While a first approach was to sample poses with only one finger contracted at a time, the insight was gained that especially for coupled motions such a model does not generalize well. In terms of posing this means that fingertips of opposing fingers do not touch or align for example. Adding specific poses that include coupled tendon contractions, as shown in the trained poses of Figure 7.9 can significantly increase generalization. This suggests that it is necessary to use poses that are related to the task that needs to be executed. Using just three additional task-specific poses (shown in Figure 7.9 on the left) the learned mapping was also precise enough to perform telemanipulation tasks, including grasping objects and inhand manipulations. Demonstrations are shown in Figure 7.10 (Left). Since the described sampling process relies exclusively on the person wearing the CyberGlove to match the robot poses with their hand, this approach may be influenced by subjective impression of how well poses match. A strong advantage of this technique however is the possibility to easily create mappings between the human hand and different hand morphologies. Given that the human operator can create a corresponding hand pose for each robot hand pose, this technique can even be applied to non-anthropomorphic foam hands.

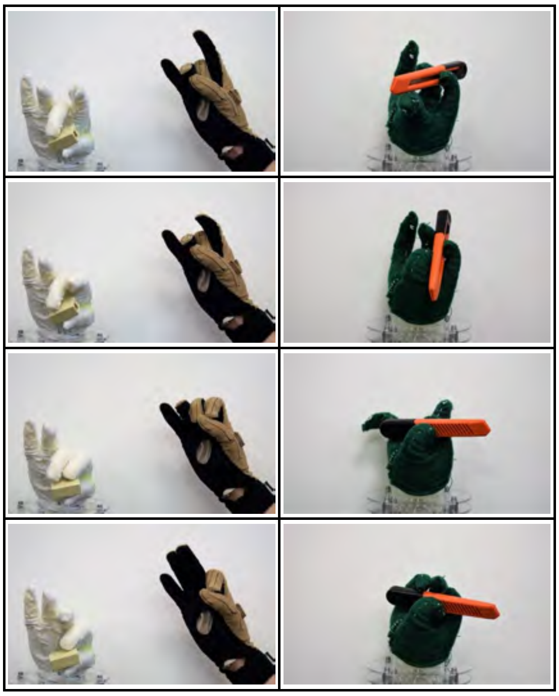


Figure 7.10.: Left) Telemanipulation sequence of a small cuboid executed by a human operator using the CyberGlove and the trained regression model. Right) Open-loop controlled manipulation sequence created by interpolating between tendon actuation keyframes.

7.5. Learning in Simulation: Inverse Kinematics

In this section, different learning-based approaches to learn a mapping between fingertip position and tendon activations are tested and evaluated in simulation. This is joint work with Kai-Hung Chang[5].

Learning on the physical robot is a straightforward and successful approach. However, the amount of test data that can be collected is limited and similarity in poses is only qualitative and depends on the patience, care, and point of view of the user. If a mapping from poses to actuations can be learned in simulation, the comparison between test poses and learned poses can be much more exact, and it can be explored how additional data may improve the results. However, for this approach to be effective, the simulation must be a good match to the actual robot hand. The accuracy of our FEM simulation for foam hands is therefore evaluated in Section 6.3.

Collecting data in simulation is faster and easier than collecting data on the physical robot. Making use of the accessibility of large amount of data from the simulation, we are able to apply learning-based methods with complex models. These methods take the concatenated fingertip positions as the input and output the tendon activation that is expected to pose the hand correspondingly. Four different methods are applied and compared:

1. Nearest neighbor
2. Linear ridge regression
3. Neural network using supervised learning
4. Deep reinforcement learning

7.5.1. Learning-based methods

The Nearest neighbor method serves as a straw-man approach. It takes the tendon activation of the pose that is nearest to the desired pose in the pose space based on Euclidean distance and simply returns that tendon activation as the result. Linear ridge regression is supplied for comparison with the experiments on the real robot. It is perhaps the second simplest sensible approach beyond Nearest Neighbor. We use a linear model with additional L2 ridge regularizer. A neural network using supervised learning adds additional degrees of freedom and nonlinearity. We include this model to determine whether the additional complexity can improve fit to the data. Our Neural Network model is constructed with four intermediate layers, each of which has 30 units and ReLU non-linear activations. The activation of the output layer is tanh(x) to match a linear-normalized range [-1,1] of the output activation. The training process runs 300 epochs with batch size 20 and Adam optimizer. Deep reinforcement learning can be considered as an alternative approach to learning a nonlinear model. Based on the success of learning IK on both rigid robot arms and hands [85], deep reinforcement learning is expected to transfer to soft robots. In particular, we apply deep deterministic policy gradient [65] algorithm combined with hindsight experience replay [1]. The shaped reward function is the negative of the average distance error over all fingers. Hindsight experience replay can be considered as a way to include additional targeted results, as ”failed” solutions are reinterpreted during learning as successful solutions to a different problem.

7.5.2. Experiments

A simulation model of the physical anthropomorphic foam hand shown in Figure 5.3 a is obtained by using Autodesk ReMake[6] to generate a surface mesh from approximately 50 images of the hand taken with a smartphone. We then run TetGen[7] to build a 3D finite element mesh of the hand. To compare the sample efficiency of all four methods, we use the same datasets for both training and testing. The training dataset collects 100,000 poses while the testing dataset contains 100 poses, all of which are pre-generated in the simulation by drawing randomly from possible tendon activations.

[5] Experiments and evaluation by Kai-Hung Chang.

[6] Autodesk remake, `https://www.autodesk.com/products/remake/overview`

[7] Tetgen, `http://wias-berlin.de/software/tetgen/`

7.5.3. Results and Discussion

We plot the performance (average distance error in centimeter) with respect to the amount of data used in training. The comparison plot is shown in Figure 7.11. When training with less than 100,000 samples, the training data is extracted in sequence from the large 100,000 dataset. The plot shows that linear ridge regression is outperformed by all other approaches especially for large datasets, implying that additional model complexity is useful for this test dataset. Overall, and to our surprise, the nearest neighbor method shows the best performance and the best sample efficiency. However, results from nearest neighbor approaches are typically not smooth for datasets that do not comprehensively cover the space of tendon actuations. Lastly, deep reinforcement learning outperforms supervised learning. The main difference between these two approaches is the existence of a loss function. While in supervised learning, the network is trained to fit the tendon activations from the training data, the objective in reinforcement learning is to maximize rewards based on the calculated average distance error, which may be physically more reasonable. Another possible cause is that the reinforcement learning algorithm, DDPG, has an actor-critic mechanism which might help the learning.

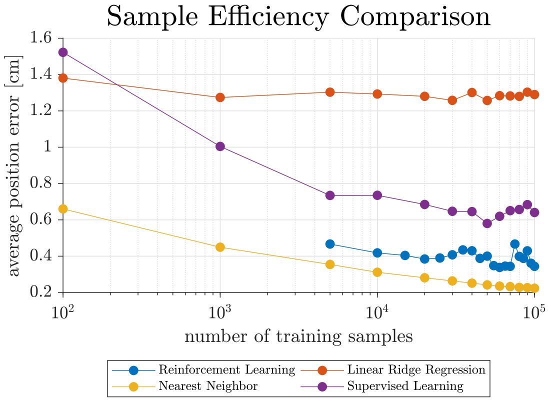


Figure 7.11.: Performance of four different methods to learn IK in simulation.[5]

8. Conclusion

8.1. Summary of Results

This work presented, evaluated and compared methods for creating and operating tendon-driven soft foam robot hands. Herein, the overall system has been considered, including fabrication, design and operation. Methods and techniques for each domain were extended, newly developed, tested and evaluated.

An improved and extended fabrication process using 3D sculpting tools and stereolithography 3D printing was proposed and several foam hands have been fabricated. Apart from a flat human hand model, non-anthropomorphic grippers were shown to be easily fabricated such as a four fingered hand and a two fingered gripper.

A modular platform was designed that contains all rigid parts such as the servo motors, and allows rapid switching between hand designs by simply detaching tendons, and reattaching and reconnecting tendons of a different hand or gripper. Effects of tendon routing, hand morphology and rest pose on achievable motions were presented and discussed. Since all three aspects influence the motion capabilities of a hand, they must be carefully selected.

An intuitive tool to explore hand motions before fabrication is the proposed simulation framework, which allows a user to load simulation meshes, create and modify tendon routings and thereby explore the hand workspace. This simulation was evaluated in terms of accuracy of predicted poses. A motion capture experiment using Vicon cameras and a four fingered gripper was conducted to obtain trajectories of fingertips while tendons were contracted. The collected position data of fingertips was compared to the positions predicted by the simulation, and an average error of 0.626cm was achieved. Possible reasons for the position error, such as friction between components of the physical system and inaccuracies between virtual and physical tendon routings were presented and discussed. The developed simulation framework was applied to conduct a user study on human ability to design tendon routings to achieve desired hand poses.

Once a foam robot hand is designed and fabricated, suitable control approaches are necessary to operate the hand. For this purpose, a modular ROS control software was created, which can be used to control a foam hand from the simulation framework or by using a CyberGlove. In operation, the foam hands demonstrates sub-millimeter repeatability and grasp strength comparable to state of the art soft robotic hands. An intuitive control approach using the CyberGlove and a regression model were shown to achieve versatile posing and manipulation with an anthropomorphic foam hand. Due to the training routine, this approach can potentially be used with arbitrarily shaped hands. Lastly, different data-driven methods to learn the inverse kinematics of a foam hand in simulation were compared.

Overall, using the developed system, tendon-driven soft foam robot hands are easily fabricated, custom designed and intuitively controlled. Their accessibility, combined with the demonstrated grasping and manipulation abilities shows that this new type of robot has great potential for robust usage in a variety of applications.

8.2. Future Work

8.2.1. Materials and Fabrication

The materials currently used for the hand prototypes are extremely flexible and enable robust grasping of objects, as shown in many demonstrations. The fabrication process could be further simplified by using 3D printed foams, where also tendon routing waypoints could be embedded to tranfer virtual tendon routings more accurately from simulation to the physical system. To increase grasp strength and thus enable pressing or pushing tasks, future work could explore embedded structures such as skeletons or variable stiffness compartments.

8.2.2. Sensors

Sensors could be used to close the control loop, as an example camera based feedback could estimate the hand pose from e.g. stereo camera images. A further alternative is installing bend sensors on the hand. This is especially interesting for using the hand in teleoperation tasks: Two identical sensorized gloves could be used to compare current hand pose of a human hand and the foam hand. However, this is only applicable in the case of an anthropomorphic foam hand, and the different morphology and different deformation behavior have to be considered. For instance, human fingers bend at defined points (joints), while foam fingers represent a continuous structure, and 'curl' up when actuated.

8.2.3. Closed-loop Control

A major motivation for the usage of soft robotic hands is that due to the high compliance of the structure itself, no low level controller is necessary to perform grasping and manipulations. The robot must however be controlled on a high level, which requires some form of feedback about the current state of the hand or grasped object. The modular ROS software developed in this work can easily be extended by such a controller module, and future work can test and evaluate the performance of different control approaches.

8.2.4. Contact Simulation

Simulating the interaction of a hand with objects requires the modeling of contacts. An attempt to implement a contact model was started in this work. Herein, collisions are detected using shape primitives (e.g. a sphere, capsule, cuboid). The objects are modeled as rigid objects using an already implemented class, where the state of an object is described by the center of mass position (3D vector), orientation (quaternion), velocity (3D vector) and angular velocity (3D vector).

A contact between a hand mesh and a rigid body is detected in the following way: For a given hand and object pose, the simulation mesh deformation and the rigid body pose are propagated into the next time step by solving the corresponding equations of motion.

If the resulting 3D position of a node of the simulation mesh is located inside the shape primitive, the closest point to this node on the boundary of the shape primitive is found. The node is then pinned to this point, and a force normal to the shape primitive surface in the contact point is attached to both the object and the node. To include a friction term, the velocity of the boundary point on the collision primitive is computed. If the velocity has a tangential component, a dynamic friction force proportional to the tangential velocity and a friction constant μ is added to both the object and the node.

An exemplary sequence of images of a three fingered hand manipulating a ball is shown in Figure 8.1. Implementing a contact simulation represents a difficult challenge and various problems must be addressed:

- It requires solving dynamic equations, which makes the simulation very slow.
- The current solver does not include a damping term (only numerical damping occurs), causing the mesh to oscillate after a change of state e.g. a tendon is contracted.

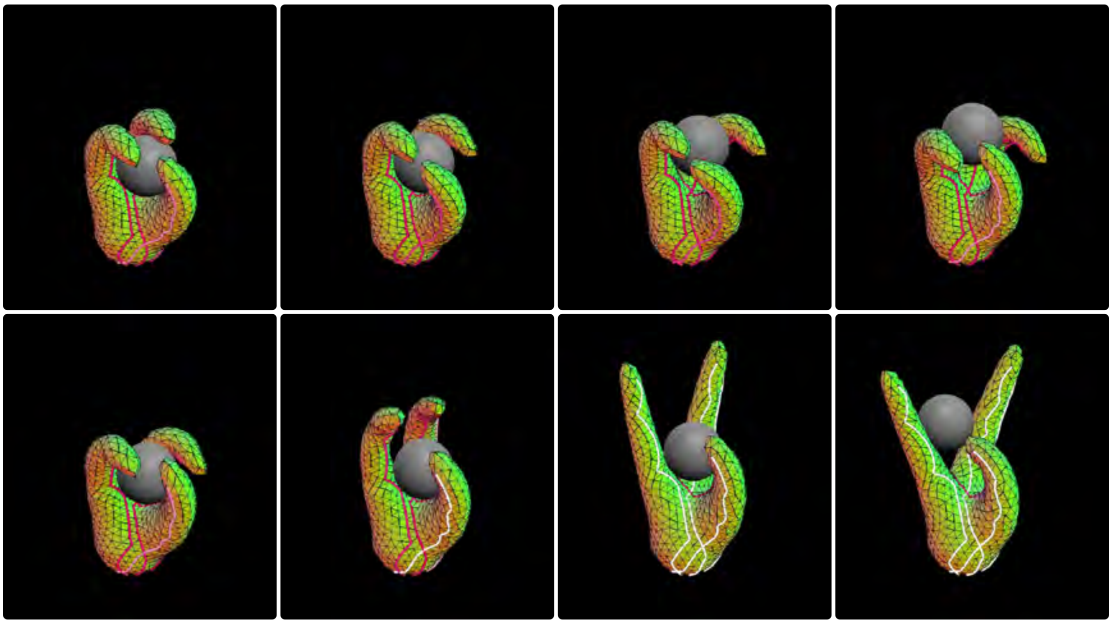


Figure 8.1.: Contact simulation. From top left to bottom right: A simulated three fingered hand manipulates a ball.

- In terms of tendon contraction, the maximum contraction speed (contraction change per simulation time step) must be limited to a realistic value.

In the attempted contact model, setting the simulation time step to a value where real-time interaction with tendons is still possible led to instabilities. In particular, larger time steps lead to collisions being detected late such that large contact forces are generated, leading to unrealistic accelerations of the object which then starts to move or spin uncontrollably. Solving these problems and implementing a realistic soft contact model is a difficult problem to be addressed in future works.

8.2.5. Learning Manipulations

If a realistic contact simulation model is implemented, learning techniques such as deep reinforcement learning could be applied and explored to learn object manipulations in simulation. Since reinforcement learning requires large amounts of data samples, it is very difficult to use this technique directly on the physical robot, as it would require a lot of time and lead to robot deterioration or failure from wear. Instead, the training can be executed in simulation, if a sufficient contact model is available.

A. Appendix

A.1. Principal Component Analysis

Principal component analysis (PCA) is a common statistical technique to detect patterns in high-dimensional data. This technique can be applied to select, transform and extract features of a dataset in order to reduce data dimensionality while minimizing information loss [76]. Dimension reduction is very helpful for problems where storage and computing resources are limited and features are redundant. Additionally, reducing the dimensions of a dataset to 2D or 3D enables visualization of the data.
PCA can be divided into the following steps:

1. A (N×s) data set $\mathbf{X}$ with N: number of data samples, s: number of features is obtained.

2. For each dimension of the data, the mean is calculated and subtracted. This yields a centered matrix.

3. The covariance matrix $\mathbf{\Sigma}$ is computed.

4. (Normalized) eigenvectors $\mathbf{v_i}$ and eigenvalues λ_i of the covariance matrix $\mathbf{\Sigma}$ are calculated. The eigenvector corresponding to the highest eigenvalue is called *principal component*. Eigenvectors are concatenated to build a matrix $\mathbf{A}$.

5. Eigenvectors in $\mathbf{A}$ are rearranged with respect to corresponding eigenvalues, in a descending sequence. Higher eigenvalues represent a stronger contribution, or higher importance, to the problem. To reduce data dimensionality, only a subset of eigenvectors is selected. The resulting matrix represents a set of orthonormal directions that point in directions of decreasing variance.

6. The data set is projected into a new (lower dimensional) feature space: $\mathbf{X}^* = \mathbf{XA}$

For a more detailed and practical tutorial on PCA, see [108].

A.2. A* Path Planning

The A* path planning algorithm is one of the most widely used approaches to find the shortest path through a network of nodes, from an initial node to a destination node. It is an iterative search algorithm which minimizes the cost

$$f(x) = g(x) + h(x) \tag{A.1}$$

where x is the current node on the path, $g(x)$ the cost of the path from the start node to the current node x, and $h(x)$ a heuristic function estimating the minimal cost from the current node to the goal node. Usually $h(x)$ is the euclidean distance between the current node and the goal. A* terminates if the path from start to goal is found or if no paths are left to be extended. The process is best described by the pseudo-code found in Algorithm 1.

Algorithm 1 A* Path Planning

```
function RECONSTRUCT_PATH(cameFrom, current)
    totalpath := current
    while current in cameFrom.Keys do
        current := cameFrom[current]
        totalpath.append(current)
    return totalpath
function A_STAR(start, goal)
    closedSet := { }
    openSet := {start}
    cameFrom := empty map
    gScore := map with default value of Infinity
    gScore[start] := 0
    fScore := map with default value of Infinity
    fScore[start] := heuristic_cost_estimate(start, goal)
    while openSet $\neq \emptyset$ do
        current := node in openSet with the lowest fScore[] value
        if current = goal then
            return reconstruct_path(cameFrom, current)
        openSet.Remove(current)
        closedSet.Add(current)
        for each neighbor of current do
            if neighbor in closedSet then
                continue                                          ▷ neighbor is already evaluated
            temp_gScore := gScore[current] + distance(current, neighbor)    ▷ distance from the start
            if neighbor not in openSet then
                openSet.Add(neighbor)
            else if temp_gScore >= gScore[neighbor] then
                continue                                          ▷ not a better path
            cameFrom[neighbor] := current
            gScore[neighbor] := temp_gScore
            fScore[neighbor] := gScore[neighbor] + heuristic_cost_estimate(neighbor, goal)
```

A.3. User Study

A.3.1. Lateral Tripod

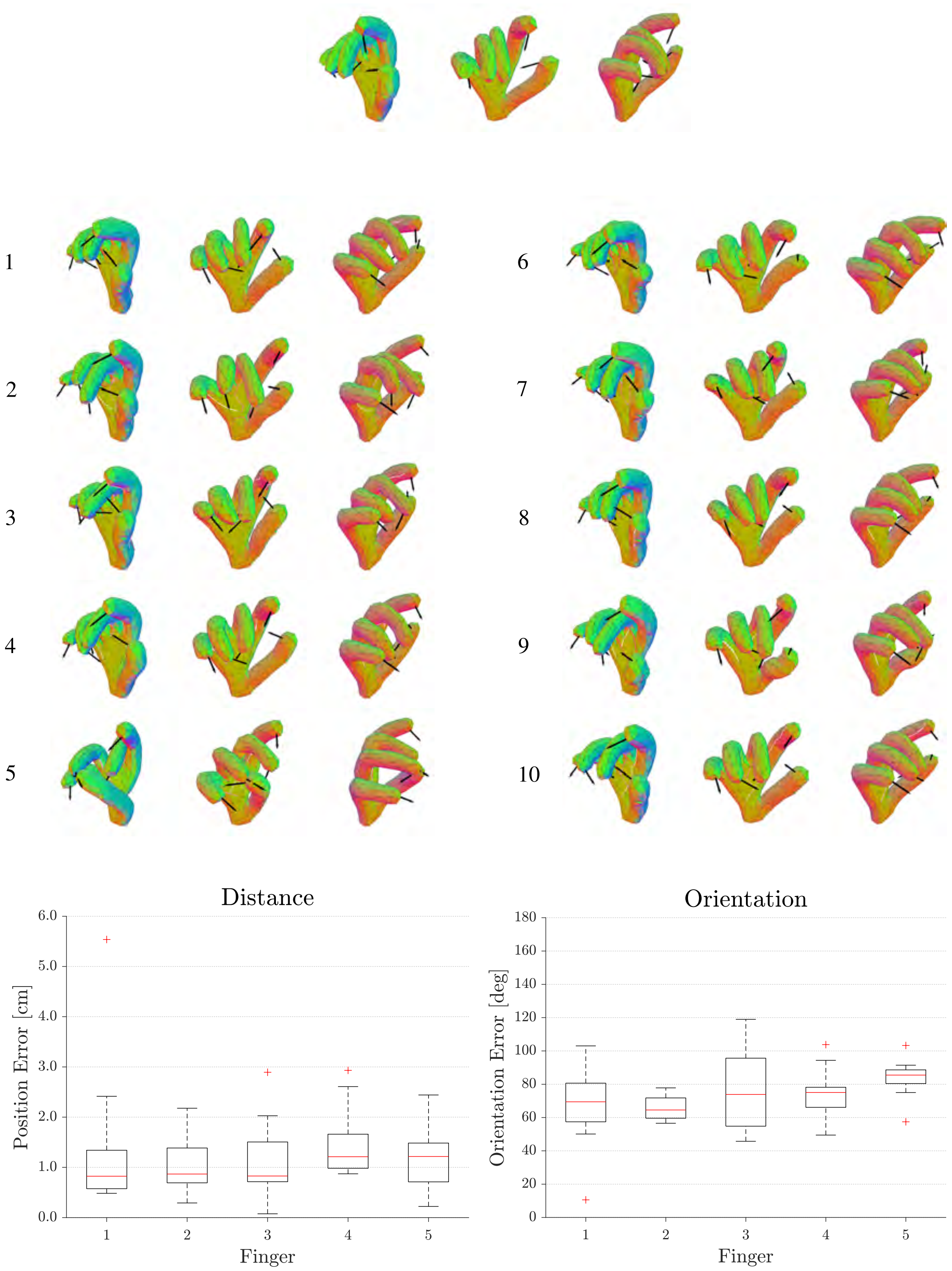


Figure A.1.: Tendon routings created for a lateral tripod grasp by 10 users in the study. Top: Target pose. Middle: Poses created by test subjects. Bottom: Position and Orientation error for each fingertip.

A.3.2. Medium Wrap a

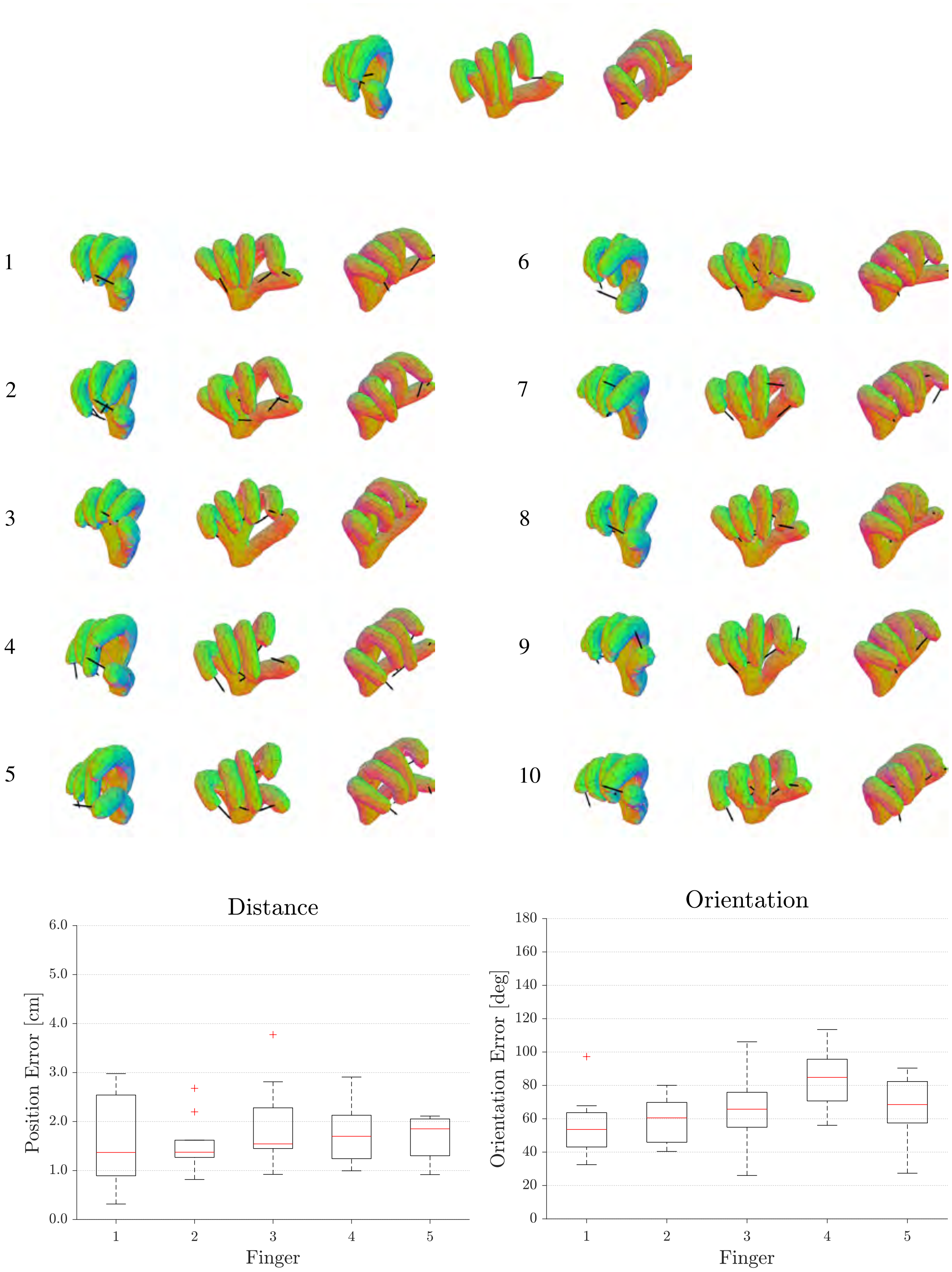


Figure A.2.: Tendon routings created for a medium wrap grasp by 10 users in the study. Top: Target pose. Middle: Poses created by test subjects. Bottom: Position and Orientation error for each fingertip.

A.3.3. Prismatic 3-finger

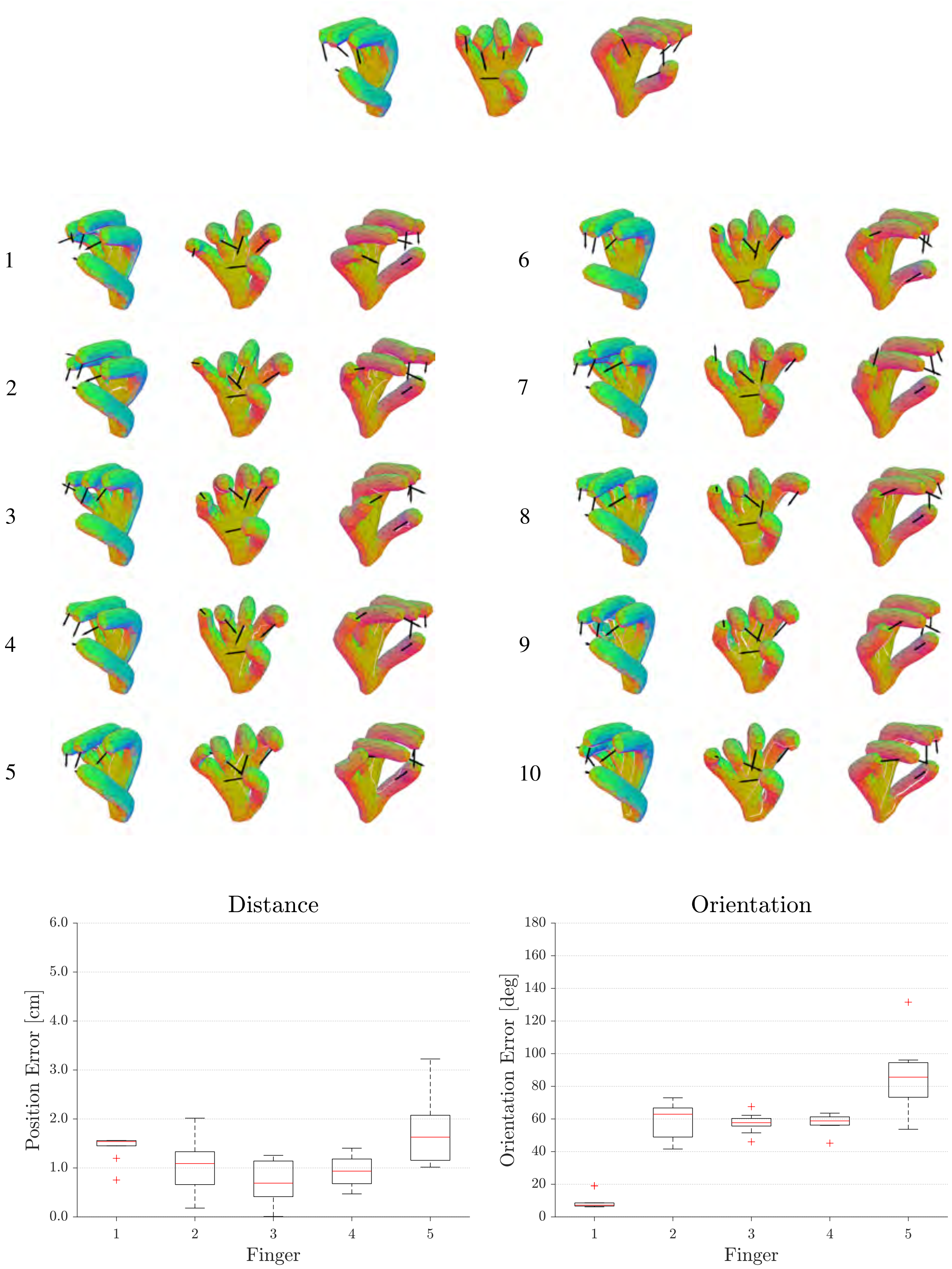


Figure A.3.: Tendon routings created for a prismatic 3-finger grasp by 10 users in the study. Top: Target pose. Middle: Poses created by test subjects. Bottom: Position and Orientation error for each fingertip.

A.3.4. Average error over all grasps

Table A.1.: Error statistics for user study. Average and maximum error over all grasps and all fingers.

Values in [cm] and [deg]	Lateral Tripod	Medium Wrap	Prismatic 3-finger
Mean distance error	1.271	1.693	1.171
Max distance error	5.538	3.773	3.225
Mean orientation error	73.723	66.297	53.918
Max orientation error	118.97	113.46	131.55

Bibliography

[1] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017. 51

[2] 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. 19

[3] D. Bauer. Automated design of tendon-driven soft foam hands using Markov-Chain-Monte-Carlo optimization methods. Master's thesis, Karlsruhe Institute of Technology, 2018. 10, 32, 39, 40

[4] G. A. Bekey, R. Tomovic, and I. Zeljkovic. *Control Architecture for the Belgrade/USC Hand*, pages 136–149. Springer New York, New York, NY, 1990. ISBN 978-1-4613-8974-3. doi: 10.1007/978-1-4613-8974-3_7. URL `https://doi.org/10.1007/978-1-4613-8974-3_7`. 18

[5] J. M. Bern, K.-H. Chang, and S. Coros. Interactive design of animated plushies. *ACM Transactions on Graphics (TOG)*, 36(4):80, 2017. 15, 22, 32, 33

[6] J. M. Bern, G. Kumagai, and S. Coros. Fabrication, modeling, and control of plush robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3739–3746, Sept 2017. doi: 10.1109/IROS.2017.8206223. 20, 22, 33

[7] A. Bicchi. Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662, Dec 2000. ISSN 1042-296X. doi: 10.1109/70.897777. 6

[8] A. Bicchi, M. Gabiccini, and M. Santello. Modelling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 366(1581):3153–3161, 2011. ISSN 0962-8436. doi: 10.1098/rstb.2011.0152. URL `http://rstb.royalsocietypublishing.org/content/366/1581/3153`. 11, 12

[9] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738. 15

[10] C. Y. Brown and H. H. Asada. Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2877–2882, Oct 2007. doi: 10.1109/IROS.2007.4399547. 12

[11] 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`. 19

[12] 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. 11

[13] M. Buss, H. Hashimoto, and J. B. Moore. Dextrous hand grasping force optimization. *IEEE Transactions on Robotics and Automation*, 12(3):406–418, 1996. 6

[14] 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. 18

[15] U. Castiello. The neuroscience of grasping. *Nature Reviews Neuroscience*, 6:726–736, 2005. 11

[16] J. G. Cham, S. A. Bailey, J. E. Clark, R. J. Full, and M. R. Cutkosky. Fast and robust: Hexapedal robots via shape deposition manufacturing. *The International Journal of Robotics Research*, 21 (10-11):869–882, 2002. 21

[17] L. Y. Chang and N. S. Pollard. Video survey of pre-grasp interactions in natural hand activities. 2009. 11

[18] 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. 10

[19] 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. 23

[20] C.-P. Chou and B. Hannaford. Measurement and modeling of mckibben pneumatic artificial muscles. *IEEE Transactions on Robotics and Automation*, 12(1):90–102, Feb 1996. ISSN 1042-296X. doi: 10.1109/70.481753. 21

[21] M. Cianchetti, A. Arienti, M. Follador, B. Mazzolai, P. Dario, and C. Laschi. Design concept and validation of a robotic arm inspired by the octopus. *Materials Science and Engineering: C*, 31(6):1230 – 1239, 2011. ISSN 0928-4931. doi: https://doi.org/10.1016/j.msec.2010.12.004. URL `http://www.sciencedirect.com/science/article/pii/S0928493110003450`. Principles and Development of Bio-Inspired Materials. 20

[22] M. Ciocarlie, C. Goldfeder, and P. Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3270–3275, Oct 2007. doi: 10.1109/IROS.2007.4399227. 12

[23] S. Cobos, M. Ferre, M. S. Uran, J. Ortego, and C. Pena. Efficient human hand kinematics for manipulation tasks. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2246–2251. IEEE, 2008. 6

[24] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, and C. Duriez. Software toolkit for modeling, simulation, and control of soft robots. *Advanced Robotics*, 31(22):1208–1224, 2017. doi: 10.1080/01691864.2017.1395362. URL `https://doi.org/10.1080/01691864.2017.1395362`. 22

[25] J. J. Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2005. 5

[26] M. R. Cutkosky. Robotic grasping and fine manipulation. 1985. 7

[27] 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, 1989. 6, 8

[28] *CyberGlove Data Glove User Guide*. CyberGlove Systems, 2007. 14

[29] 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. doi: 10.1177/0278364915592961. URL `https://doi.org/10.1177/0278364915592961`. 20, 21

[30] 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. 21, 24

[31] T. Doll and H.-J. Schneebeli. The karlsruhe hand. *IFAC Proceedings Volumes*, 21(16):383 – 388, 1988. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)54640-6. URL `http://www.sciencedirect.com/science/article/pii/S1474667017546406`. 2nd IFAC Symposium on Robot Control 1988 (SYROCO '88), Karlsruhe, FRG, 5-7 October. 18

[32] 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. 23

[33] S. Edwards, D. Buckland, and J. McCoy-Powlen. Developmental & functional hand grasps. 2002. *Slack Incorporated, Thorofare, New Jersey*. 10

[34] P. Eichelberger, M. Ferraro, U. Minder, T. Denton, A. Blasimann, F. Krause, and H. Baur. Analysis of accuracy in optical motion capture–a protocol for laboratory setup evaluation. 49, 05 2016. 13

[35] 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`. 10, 11

[36] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016. 7, 8, 21

[37] 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. 19

[38] 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. 19

[39] 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. 19

[40] 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. 23

[41] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, et al. The dlr hand arm system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3175–3182. IEEE, 2011. 18

[42] G. S. Guthart and J. K. Salisbury. The intuitive/sup tm/ telesurgery system: overview and application. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 618–621 vol.1, April 2000. doi: 10.1109/ROBOT.2000.844121. 23

[43] J. Hiller and H. Lipson. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466, 2012. 21, 22, 24

[44] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. ISSN 00401706. URL http://www.jstor.org/stable/1267351. 16

[45] W. S. Howard and V. Kumar. On the stability of grasped objects. *IEEE transactions on robotics and automation*, 12(6):904–917, 1996. 7

[46] T. Iberall. Grasp planning from human prehension. In *IJCAI*, volume 87, pages 1153–1157. Citeseer, 1987. 7

[47] T. Iberall. The nature of human prehension: Three dextrous hands in one. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 396–401. IEEE, 1987. 6

[48] T. Iberall. Human prehension and dexterous robot hands. *The International Journal of Robotics Research*, 16(3):285–299, 1997. 7

[49] 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. 20, 21, 22

[50] 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. 21, 24

[51] 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. 18

[52] M. Jeannerod. Intersegmental coodination during reaching at natural visual objects. *Attention and performance*, pages 153–169, 1981. 11

[53] M. Jeannerod. The timing of natural prehension movements. *Journal of Motor Behavior*, 16 (3):235–254, 1984. doi: 10.1080/00222895.1984.10735319. URL https://doi.org/10.1080/00222895.1984.10735319. PMID: 15151851. 11

[54] B. A. Jones and I. D. Walker. Kinematics for multisection continuum robots. *IEEE Transactions on Robotics*, 22(1):43–55, Feb 2006. ISSN 1552-3098. doi: 10.1109/TRO.2005.861458. 22

[55] 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. 8, 10

[56] S. B. Kang and K. Ikeuchi. A framework for recognizing grasps. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1991. 10

[57] I. Kapandji. *The Physiology of the Joints: Upper limb*. The Physiology of the Joints: Annotated Diagrams of the Mechanics of the Human Joints. E. & S. Livingstone, 1970. URL https://books.google.de/books?id=cYhIuAEACAAJ. 8

[58] I. A. Kapandji. *Funktionelle Anatomie der Gelenke: schematisierte und kommentierte Zeichnungen zur menschlichen Biomechanik*. Georg Thieme Verlag, 2009. 10

[59] 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, Sept 2002. ISSN 1083-4435. doi: 10.1109/TMECH.2002.802720. 18

[60] J. Kerr and B. Roth. Analysis of multifingered hands. *The International Journal of Robotics Research*, 4(4):3–17, 1986. 6

[61] 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`. 20

[62] A. Kochan. Shadow delivers first hand. *Industrial Robot: An International Journal*, 32(1): 15–16, 2005. doi: 10.1108/01439910510573237. URL `https://doi.org/10.1108/01439910510573237`. 18

[63] C. Kozierok. *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. No Starch Press, San Francisco, CA, USA, 2005. ISBN 159327047X, 9781593270476. 12

[64] J. J. Kuch and T. S. Huang. Human computer interaction via the human hand: a hand model. In *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, volume 2, pages 1252–1256. IEEE, 1994. 6

[65] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 51

[66] 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. 29

[67] H.-T. Lin, G. G. Leisk, and B. Trimmer. Goqbot: a caterpillar-inspired soft-bodied rolling robot. *Bioinspiration & Biomimetics*, 6(2):026007, 2011. URL `http://stacks.iop.org/1748-3190/6/i=2/a=026007`. 20, 21, 22

[68] H. Liu, T. Iberall, and G. A. Bekey. The multi-dimensional quality of task requirements for dextrous robot hand control. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 452–457. IEEE, 1989. 6

[69] C. S. Lovchik and M. A. Diftler. The robonaut hand: a dexterous robot hand for space. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 2, pages 907–912 vol.2, May 1999. doi: 10.1109/ROBOT.1999.772420. 18

[70] C. L. MacKenzie and T. Iberall. *The grasping hand*, volume 104. Elsevier, 1994. 7

[71] 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`. 23

[72] A. D. Marchese, K. Komorowski, C. D. Onal, and D. Rus. Design and control of a soft and continuously deformable 2d robotic manipulation system. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2189–2196, 2014. 23

[73] L. Margheri, C. Laschi, and B. Mazzolai. Soft robotic arm inspired by the octopus: I. from biological functions to artificial requirements. *Bioinspiration & Biomimetics*, 7(2):025004, 2012. URL `http://stacks.iop.org/1748-3190/7/i=2/a=025004`. 22

[74] J. McCann, L. Albaugh, V. Narayanan, A. Grow, W. Matusik, J. Mankoff, and J. Hodgins. A compiler for 3d machine knitting. *ACM Trans. Graph.*, 35(4):49:1–49:11, July 2016. ISSN 0730-0301. doi: 10.1145/2897824.2925940. URL `http://doi.acm.org/10.1145/2897824.2925940`. 28, 38

[75] C. Melchiorri and M. Kaneko. Robot hands. In *Springer Handbook of Robotics*, pages 463–480. Springer, 2016. 18

[76] R. Mikut. Data mining in der medizin und medizintechnik. Technical report, Karlsruher Institut für Technologie (KIT), 2008. Erweitere Ausg. d. Habilitationsschrift u.d.T.: Automatisierte Datenanalyse in der Medizin und Medizintechnik, 2007. 56

[77] R. M. Murray. *A mathematical introduction to robotic manipulation*. CRC press, 2017. 4, 5, 7

[78] 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. 20, 21, 22

[79] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 16

[80] J. R. Napier. The prehensile movements of the human hand. *The Journal of bone and joint surgery. British volume*, 38(4):902–913, 1956. 8

[81] 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. 26

[82] T. Okada. Object-handling system for manual industry. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(2):79–89, 1979. 18

[83] D. Osswald, J. Martin, C. Burghart, R. Mikut, H. Wörn, and G. Bretthauer. Integrating a flexible anthropomorphic, robot hand into the control, system of a humanoid robot. *Robotics and Autonomous Systems*, 48(4):213 – 221, 2004. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2004.07.005. URL `http://www.sciencedirect.com/science/article/pii/S0921889004000983`. Humanoids 2003. 18

[84] A. Pfister, A. M. West, S. Bronner, and J. A. Noah. Comparative abilities of microsoft kinect and vicon 3d motion capture for gait analysis. *Journal of Medical Engineering & Technology*, 38(5): 274–280, 2014. doi: 10.3109/03091902.2014.909540. URL `https://doi.org/10.3109/03091902.2014.909540`. 13

[85] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018. 51

[86] 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*, pages 311–318. ACM, 2005. 6

[87] D. Prattichizzo and J. C. Trinkle. Grasping. In *Springer handbook of robotics*, pages 671–700. Springer, 2008. 7

[88] M. Rakić. Multifingered robot hand with selfadaptability. *Robotics and Computer-Integrated Manufacturing*, 5(2):269–276, 1989. ISSN 0736-5845. doi: https://doi.org/10.1016/0736-5845(89)90074-4. URL `http://www.sciencedirect.com/science/article/pii/0736584589900744`. 18

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

[90] 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. 21, 24

[91] 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`. 10

[92] 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, 2014. 23

[93] J. Romero. jrgn@kth.se. 10

[94] D. Rus and M. T. Tolley. Design, fabrication and control of soft robots. *Nature*, 521:467–475, 2015. 19, 20, 21, 22

[95] 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. 7, 18

[96] M. Santello, M. Flanders, and J. F. Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.18-23-10105.1998. URL `http://www.jneurosci.org/content/18/23/10105`. 11

[97] 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`. 23

[98] S. Schulz, C. Pylatiuk, and G. Bretthauer. A new class of flexible fluidic actuators and their applications in medical engineering. 47:390–395, 01 1999. 18

[99] S. Schulz, C. Pylatiuk, and G. Bretthauer. A new ultralight anthropomorphic hand. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 3, pages 2437–2441 vol.3, May 2001. doi: 10.1109/ROBOT.2001.932988. 18

[100] S. Schulz, C. Pylatiuk, M. Reischl, J. Martin, R. Mikut, and G. Bretthauer. A hydraulically driven multifunctional prosthetic hand. 23:293–299, 05 2005. 18, 19

[101] Y. Seol, C. O'Sullivan, and J. Lee. Creature features: Online motion puppetry for non-human characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 213–221, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2132-7. doi: 10.1145/2485895.2485903. URL `http://doi.acm.org/10.1145/2485895.2485903`. 23, 50

[102] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014. ISBN 1107057132, 9781107057135. 15, 17

[103] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521813972. 16

[104] 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. ISSN 0027-8424. doi: 10.1073/pnas.1116564108. URL `http://www.pnas.org/content/108/51/20400`. 20, 21

[105] K. B. Shimoga. Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, 15(3):230–266, 1996. 6, 7

[106] 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`. 19, 20, 21

[107] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2016. 4, 5, 23

[108] L. I. Smith. A tutorial on principal components analysis. Technical report, 2002. URL `http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf`. 56

[109] B. Stenger, P. R. Mendonça, and R. Cipolla. Model-based 3d tracking of an articulated hand. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2001. 6

[110] R. Sutton and A. Barto. Reinforcement learning: An introduction, (complete draft), 2017. 17

[111] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981. 17

[112] Ö. Terlemez, S. Ulbrich, C. Mandery, M. Do, N. Vahrenkamp, and T. Asfour. Master motor map (mmm)—framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 894–901. IEEE, 2014. 6

[113] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi. Control strategies for soft robotic manipulators: A survey. *Soft robotics*, 5(2):149–163, 2018. 5

[114] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi. Control strategies for soft robotic manipulators: A survey. *Soft robotics*, 5 2:149–163, 2018. 22, 23

[115] M. T. Tolley, R. F. Shepherd, B. Mosadegh, K. C. Galloway, M. Wehner, M. Karpelson, R. J. Wood, and G. M. Whitesides. A resilient, untethered soft robot. *Soft Robotics*, 1(3):213–223, 2014. doi: 10.1089/soro.2014.0008. URL `https://doi.org/10.1089/soro.2014.0008`. 20, 21

[116] W. Townsend. The barretthand grasper – programmably flexible part handling and assembly. *Industrial Robot: An International Journal*, 27(3):181–188, 2000. doi: 10.1108/01439910010371597. URL `https://doi.org/10.1108/01439910010371597`. 18

[117] R. J. Webster and B. A. Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *I. J. Robotics Res.*, 29:1661–1683, 2010. 22

[118] Y. Xia and G. M. Whitesides. Soft lithography. *Annu. Rev. Mater. Sci*, 28:153–84, 1998. 22

[119] 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. 20, 23

[120] 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(GMAP)*, volume 00, pages 57,58,59,60,61,62,63,64,65,66,67,68, 2000. doi: 10.1109/GMAP.2000.838238. URL `doi.ieeecomputersociety.org/10.1109/GMAP.2000.838238`. 29