DESIGN OF A TENDON DRIVEN SOFT CONTINUUM ROBOTIC ARM USING PROPRIOCEPTIVE SENSORS

Richard Sperling
University of Rhode Island, r-sperling@my.uri.edu

Follow this and additional works at: https://digitalcommons.uri.edu/theses

Recommended Citation
https://digitalcommons.uri.edu/theses/1286

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons@etal.uri.edu.
DESIGN OF A TENDON DRIVEN SOFT CONTINUUM ROBOTIC ARM

USING PROPRIOCEPTIVE SENSORS

BY

RICHARD SPERLING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND
2018
MASTER OF SCIENCE THESIS

OF

RICHARD SPERLING

APPROVED:

Thesis Committee:

Major Professor

Hongyan Yuan

Chengzhi Yuan

Brennan Phillips

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2018
Abstract

In this thesis the use of proprioceptive sensors in elephant trunk inspired soft continuum robots was explored. After presenting the basics for this project, a review of existing soft continuum robots is given. The sensors used are off-the-shelf stretch sensors. Those sensors are carbon black impregnated rubber cords. The design and the fabrication of the robot are described. Using material characteristics a calibration process was developed to improve the sensor’s consistency. The robot is then tested in open-loop as well as a closed-loop experiment. Results of those experiments are presented and discussed.
Acknowledgment

I would first like to thank my thesis advisor Dr. Hongyan Yuan of the Department of Mechanical Engineering at the University of Rhode Island. The door to Prof. Yuan office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this project to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to acknowledge Bahador Marzban and Reed Nelson of the Department of Mechanical Engineering at the University of Rhode Island, who supported me in conducting the experiments and offered me their opinion on the current state of the project.
Contents

Abstract ii

Contents iv

List of Figures vii

List of Tables ix

List of Algorithms x

Nomenclature xi

1 Introduction 1

2 Basics 4

2.1 Kinematics 4

2.2 Micro Controller Unit 5

2.2.1 Measuring Resistance using an MCU 6

2.2.2 Interrupt Functions 8

2.2.3 Pulse Width Modulation 9

2.3 Electromagnetic Tracking Sensor 10
3 State of the Art

3.1 Continuum Robots

3.1.1 Continuous and Discrete Robots

3.1.2 Extensibility

3.1.3 Number of Sections

3.1.4 Actuators per Section

3.1.5 Degrees of Freedom per section

3.1.6 Actuator Spacing

3.1.7 Actuation

3.2 Soft Robots

4 Experimental Setup

4.1 Robot Design

4.2 Fabrication

4.3 Control

4.4 Kinematic

5 Material Characterization

6 Robot Control

6.1 Calibration

6.2 Open-Loop Control
List of Figures

2.1 Different MCUs ......................................................... 5
2.2 Ohm meter using a MCU ............................................. 7
2.3 Error of measuring Resistance ................................. 8
2.4 PWM examples ..................................................... 9
2.5 EM sensors [89] ..................................................... 10
2.6 EM field generator [89] ........................................... 11
3.1 Examples of hydroskeletons and muscular hydrostats ...... 17
3.2 Different robot designs in literature ......................... 19
4.1 Schematic of the robot ........................................... 22
4.2 robot cross section ................................................. 22
4.3 Mold used for robot fabrication ............................... 24
4.4 Setup of the electronics .......................................... 25
4.5 Variable nomenclature for kinematic ....................... 27
4.6 Space mapping ...................................................... 29
4.7 Task space .......................................................... 30
5.1 Cyclic stress- strain curve for the Ecoflex 00-30 .......... 35
5.2 Cyclic stress- strain curve for the conductive rubber cord . 36
5.3 Steady Resistance- strain curve for the conductive rubber band [2] up to 20% engineering strain, which shows significant aging effect. 37
List of Tables

3.1 Properties of robots used in literature ........................................ 13

3.2 Comparison between hydraulic/pneumatic actuation and tendon driven
actuation ................................................................. 16

4.1 Material properties of Ecoflex[117] ........................................... 23

4.2 Commands for control via serial connection with the MCU ............ 28
<table>
<thead>
<tr>
<th>Algorithm Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 speed output decision logic</td>
<td>26</td>
</tr>
<tr>
<td>2 calibration process</td>
<td>40</td>
</tr>
</tbody>
</table>
Nomenclature

$\alpha_i$  
Angle of sensors

$\delta$  
Distance of tendons/sensors from the center of the robot

$\ell$  
Length along thread in the center

$\kappa$  
Bending curvature

$q$  
Actuator space

$s$  
Sensor space

$x$  
Task space

$\phi$  
Angle to bending plane

$\theta$  
Bending angle

$\varphi_i$  
Angle of tendons

$\vec{p}$  
End-effector position in bending plane

$I$  
Electric current

$l_i$  
Tendon lengths

$N_{actuators}$  
Number of actuators

$r$  
Bending radius

$\xi$
$R_C$  Electric stretch sensor resistance

$R_{PD}$  Pull-down resistance

$s_i$  Sensor lengths

$T$  Transformation matrix

$T_\phi$  Transformation matrix for bending direction

$T_{r,\theta}$  Transformation matrix in bending plane

$V_{in}$  Input voltage

$V_{out}$  Output/measured voltage

$V_{RC}$  Voltage drop over $R_C$

$x$  x-Coordinate of end-effector

$x_{ci}$  x-Coordinate of tendons on xy-plane

$x_{si}$  x-Coordinate of sensors on xy-plane

$y$  y-Coordinate of end-effector

$y_{ci}$  y-Coordinate of tendons on xy-plane

$y_{si}$  y-Coordinate of sensors on xy-plane

$z$  z-Coordinate of end-effector

xii
1 Introduction

To mimic the locomotion and manipulation mechanisms of soft biological organisms, soft robotics have been attracting intensive research interests [109] recently. Soft robots are made primarily of elastomeric materials which undergo large and continuous deformation when subject to actuator (internal) and environmental (external) forces. As a result, soft robots can adapt their shapes to variable, geometrically complex environments [120] and are safe to operate when interacting with humans, animals, or manipulating human tissues/organs [108, 101]. The versatility and adaptability of soft robots make them better suited for open-ended tasks such as navigating in uncertain environments. Applications of soft robots as manipulators include a soft gripper for pick-and-place articulation [55], a soft contractable sleeve as heart assistive device [108], a soft arm for deep-sea exploration [100], a soft actuated system for gait rehabilitation [101]. Autonomous mobile soft robots have been designed to explore the modes of locomotion, such as a tissue-engineered soft robotic stingray with light sensing [94], a resilient walking soft robot in hazardous conditions [120], and a soft robotic fish capable of escape maneuvers [82]. While significant advances have been made in the field of soft robotics in recent years, obstacles still exist in fabrication, sensing, modeling and control of soft robots which have prevented full realization of the promising applications of soft robots.

While the compliance of soft robots is an integral part of their functions, it poses a challenge in the sensing of robotic configurations. Many traditional sensors for rigid
robots, such as motor encoders, do not work well for soft robots, especially when the soft robots physically interact with environments. This is because soft robots can undergo large continuum deformation. The kinematic relations that describe the mappings between actuation space, configuration space, and task space of soft robots not only depend on robot geometry, but also material stress-strain relationship and environmental constraints. In the general cases, nonlinear finite-strain elasticity problems need to be solved to obtain the kinematic relations [88, 35], which are computational intensive when solved numerically using the finite element method.

Both external and internal sensors have been explored to acquire the configuration of soft robots. External position tracking systems, such as electromagnetic tracking [49] and optically-based 3D motion capture [45], has been used to obtain shape sensing for feedback control. External sensing systems work well for situations where modestly-sized robots perform tasks in a predefined area which can be covered by the external tracking system. For long distance tasks or large scale robotic designs, such as autonomous mobile robots or extendable robotic arms, a means of internal sensing is preferred. The primary deformation mode of soft robotic arms is bending; therefore, to accurately predict the end-state configuration, it is required to know the curvature along the soft-robotic arm. Several proprioceptive sensors that are able to achieve this have been developed and/or implemented previously. Such sensors include: a Hall Effect sensor [92]; a piezoelectric deflection sensor [113]; a soft-strain sensor which utilizes conductive fabric [33]; fiber-optic sensors [111]. Such sensors, in principle, can measure the local curvature along the
robotic arm. A simpler approach is to assume the curvature along the length of each segment of a modular arm is constant; therefore, measuring the length along different paths in the longitudinal direction of the arm segment allows one to compute the curvature based on the geometric relations. Previous work following this approach includes a length sensor based on inductance change of conductive braids [48], soft-strain sensors which utilize either liquid or carbon nanotube conductors [53], and a commercially available bilayer thin film sensor [50].

Various actuation strategies have been used in soft robotics for locomotion and articulation; these include flexible fluidic actuators [110, 88], tendon-drives [104], shape memory alloys [127], electro-active polymers [91], etc. Among them, the tendon-driven approach, where DC motors pull tendons inside the robotic arm to generate curvature, has several advantages. First, the use of tendon-drive actuation has been extensively proven by in traditional hard continuum robotic arms. The mechanics models developed previously for a tendon-driven continuum arm can be used, modified, and extended to the soft tendon-driven robotic arms. Secondly, because of the softness and light-weight nature of soft robotic arms, small batteries and miniature motors will suffice for actuation.

In this work, we explore the curvature sensing using conductive soft filaments for the feedback control of a tendon-driven soft robotic arm. Ideally, a low-cost proprioceptive sensor made of soft materials is preferred such that it does not restrict the deformation of soft robots in any directions.
2 Basics

In this section the basic knowledge of for this project will be presented. This includes:

- the mathematical description of robots (subsection 2.1)
- the usage of MCU (Micro Controller Units) as interface between hardware and software

2 Kinematics

In robotics a common way to find an end-effector position for a robot is by using a homogeneous transformation. With a known homogeneous transformation matrix $T$ and known base coordinates $x_b$, $y_b$, and $z_b$, the end-effector coordinates $x_e$, $y_e$, and $z_e$ can be derived using Equation 2.1. $R$ describes the rotation of the base and $\vec{t}$ the translation of the base.

$$
\begin{bmatrix}
  x_e \\
  y_e \\
  z_e \\
  1 \\
\end{bmatrix} =
\begin{bmatrix}
  R & \vec{t} \\
  0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  x_b \\
  y_b \\
  z_b \\
  1 \\
\end{bmatrix}
$$

(2.1)

The transformation matrix $T$ dependent on the actuator states. For example: A tendon driven robot this with 3 tendons would have a transformation matrix that is dependent on the 3 tendon lengths $T(l_1, l_2, l_3)$. Mapping the actuator states to the end-effector position is known as forward kinematics. Mapping the end-effector coordinates to the
actuator states is called inverse kinematics and requires solving the system of equations (Equation 2.1).

2 Micro Controller Unit

MCUs (Micro Controller Unit) are small computer in an integrated circuit. On those MCUs small code (due to limited memory) can be run and specific Voltages on physical pins can be output or measured. A few examples of MCUs are shown in Figure 2.1.

![Figure 2.1: Different MCUs](image)

The main difference between the Arduino Uno (see Figure 2.1a) and the Arduino Pro Mini (see Figure 2.1b) is the size and the number of pins. Also the Arduino Pro Mini doesn’t have a USB to TTL serial converter like the Arduino Uno. The ESP-12E uses a different processor than the Arduino chips and requires only a 3.3 V power supply. However it can therefore output a maximum of 3.3 V. The ESP-12E MCU also has an ESP8266 WiFi Module.

MCUs posses 3 different types of pins:
• digital IO pins

• analog input pins

• PWM output pins (can also be used as digital IO pins)

Digital IO pins have to be set to either input or output mode. In input mode the
MCU read a high voltage if the potential at the pin is above a certain threshold () and
reads a low voltage otherwise. In output mode the MCU can set the potential at the pin
to a high voltage (5 V) or a low voltage (0 V).

To program those MCUs the Arduino IDE can be used. This software allows the
user to program using a programming language that is similar to c++. Instead of a main
function the Arduino language uses a setup function and a loop function. The setup
function is executed once, when the MCU is started (or the reset button is pressed).
After that the loop function is executed repeatedly.

2 Measuring Resistance using an MCU

To measure a unknown resistance using an Arduino board the circuit is setup as shown
in Figure 2.2a. A schematic of this circuit is shown in Figure 2.2b. Here one of the
analog input pins is used to measure the voltage drop over the pull-down resistor. The
analog inputs in the range of 0 V

In terms of this project the unknown resistor will be replaced with the sensors (conduc-
tive rubber cords). Here $V_{in}$, $R$ (pull-down resistor) and $V_{out}$ are known. The pull-down

6
resistor is required to prevent current from flowing into the analog input. The current flowing through both resistors must be equal:

\[ I = \frac{V_{\text{out}}}{R} = \frac{V_{R_C}}{R_C} \]  \hspace{1cm} (2.2)

According to Kirchhoff’s second law Equation 2.3 holds.

\[ V_{\text{in}} = V_{R_C} + V_{\text{out}} \]  \hspace{1cm} (2.3)
From those equations $R_C$ can be derived:

$$R_C = \left( \frac{V_{in}}{V_{out}} - 1 \right) R$$

(2.4)

Because a MCU doesn’t read analog voltage values, but instead converts those to digital values, we have to deal with errors during the readings. Section 2.2.1 shows the error of the measurement in dependence of the actual resistance of the unknown resistance. The error has its minimum for a resistance value similar to the pull-down resistor.

![Figure 2.3: Error of measuring Resistance](image)

### 2 Interrupt Functions

To trigger functions in through a signal applied at the input pins Arduino Boards offer interrupt functions. Whenever a event (rising voltage, falling voltage) occurs at a pin
with an interrupt function attached to it, the execution of the code is paused and the interrupt function is executed. After that the program continues where it was paused. This feature can be used to detect voltage peaks. In this project interrupt functions are used in combination with motor encoders.

2 Pulse Width Modulation

Pulse width modulation is a method to create a analog voltage by switching between 2 different voltage potentials. By changing the width of a pulse (high voltage level) without changing the width for a whole cycle a different average voltage can be achieved. Even though this technically is not a analog voltage output, in many electronic circuits it can be treated as one, because PWM usually uses a very high frequency. Figure 2.4a shows a example for a 50% duty cycle with a 0 V and a 5 V potential. The voltage output is 2.5 V. Figure 2.4b shows a 75% duty cycle with a 3.75 V output.
2 Electromagnetic Tracking Sensor

Electromagnetic tracking sensor are sensors that can be used to track a position and orientation accurately. The system consists of a field generator that creates a oscillating electromagnetic field. Sensors such as those shown in Figure 2.5 consist of small solenoids. Due to the electromagnetic field a current is induced in those solenoids. Depending on the location and orientation of the sensors different currents are induced in the solenoids. With this data the position and orientation of the sensors relative to the field generator can be tracked.

The accuracy of those sensors decreases when disruptive factors are near. Because the sensors rely on the electromagnetic field, any kind of ferromagnetic material or electric current nearby introduces a error.
Figure 2.6: EM field generator [89]
3 State of the Art

Most soft robots can also be classified as continuum robots. Therefore the following sections describe those continuum robots. After that an introduction to soft robots is given.

3 Continuum Robots

Continuum robots are robots that mainly move by deforming parts. In this section the focus will be put on elephant trunk or snake shaped robots. Those continuum robots can be compared using different criteria [115, 59, 107, 128]:

- continuous or discrete
- extensibility
- number of sections
- actuators per section
- DOF per section
- actuator spacing
- actuation

Table 3.1 shows robots used in literature with their respective criteria.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Continuous / discrete</th>
<th>Extensible</th>
<th>Number of sections</th>
<th>Actuators per section</th>
<th>DOF per section</th>
<th>Actuator spacing</th>
<th>Actuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[63, 64]</td>
<td>C</td>
<td>✓</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>180°</td>
<td>P</td>
</tr>
<tr>
<td>65</td>
<td>C (✓)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>C</td>
<td>✓</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>120°</td>
<td>P</td>
</tr>
<tr>
<td>44</td>
<td>C</td>
<td>✓</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>120°</td>
<td>P</td>
</tr>
<tr>
<td>87</td>
<td>C ×</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>C ×</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>non uniform</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>C</td>
<td>✓</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>120°</td>
<td>T</td>
</tr>
<tr>
<td>[70, 69]</td>
<td>C</td>
<td>✓</td>
<td>4</td>
<td>4</td>
<td>120°</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>C (✓)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>C ×</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>180°</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>C ✓</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>120°</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>C ×</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>T/P</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>C ✓</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>60°</td>
<td>T/P</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>C ×</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>C ✓</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>180°</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>[103, 104]</td>
<td>C</td>
<td>✓</td>
<td>4</td>
<td>3</td>
<td>90°</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>[80, 81, 83, 84]</td>
<td>C ✓</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>180°</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>C ×</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>C (✓)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>C ✓</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>90°</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>C</td>
<td>✓</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>180°</td>
<td>P</td>
</tr>
<tr>
<td>90</td>
<td>C ✓</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>120°</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>C ✓</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>120°</td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Properties of robots used in literature (extension to table in [128])
3 Continuous and Discrete Robots

A Continuous robot is one that moves by deforming parts. A discrete robot is one that moves by having parts that purely rotate or translate. However many discrete robots have a very high number of small joints. Due to the high number of joints a approximation as continuum robot can be made. If those joints can be actuated individually those robots become highly redundant as every joint adds additional DOF. [128]

3 Extensibility

The extensibility of a robot describes whether compression and lengthening of the body is taken into account or whether the robot purely bends. For a inextensible robot the number of linear independent DOF of a section is capped to 2. A extensible robot on the other hand can have up to 3 linear independent DOF. [128]

3 Number of Sections

For a robot that consists of more than one section, more DOF are added. Those additional DOF however are not necessarily linear independent and therefor can add redundancy to the whole robot. Despite having multiple sections most robots are capped in the number of sections. This is due to the fact that the actuation of one section still takes place at the base of the robot. In this case tendons or tubes for the actuation of the later sections pass through the first sections. [128]
3 Actuators per Section

The number of actuators per section is strongly related to number of DOF. A robot with only 2 actuators can only have 2 linear independent DOF. Some robot design contain 4 actuators in one section. By having more than 3 actuators a robot can achieve a bidirectional behavior. Many other robots achieve the bidirectional behavior by having springs that pull the robot in the other direction or have bidirectional actuators. [128]

3 Degrees of Freedom per section

The DOF of a robot represents how complex the movement can be. While 6 DOF would be enough to reach any point in a 3 dimensional space from any direction, it is restricted in path it takes to the point. To deal with restrictions from the environment more DOF are added (usually by adding more sections). [128]

3 Actuator Spacing

The actuator spacing describes the location of the actuators. Snake like robots usually have their actuators located symmetrically around the center. So the actuator spacing is dependent on the number of actuators \( (\varphi_{i+1} - \varphi_i = \frac{2\pi}{N_{\text{actuators}}}) \). [128]

3 Actuation

The actuation in many cases is done using tendons or hydraulic/pneumatic pumps. While it is usually easy for a pump to achieve big bending angles due to high pressure,
they bring a few disadvantages with them. Table 3.2 shows a comparison between hydraulic/pneumatic actuation and tendon driven actuation. The fact that a tendon driven robot could possibly carry its own actuation makes such a system suitable for mobile applications. [128]

### 3 Soft Robots

In soft robotics the robots are made out of a soft material such as silicon rubber instead of rigid joints and arms. Usually those robot are classified as continuous robots as their main way of moving is through elastic deformation of the soft material. Examples for inspiration from nature are shown in Figure 3.1. Pneumatic or hydraulic actuation is widely popular for those kinds of robots as the pressure chambers can easily be embedded in the robot’s body. [85, 87, 88, 51, 102, 118]

In [82] a fish like robot was developed (see Figure 3.2a), that was capable of maneuv...
Figure 3.1: Examples of hydroskeletons and muscular hydrostats: (a) tube feet in starfish, (b) octopus arms, (c) colonial anemone, (d) mammalian tongue, (e) squid, (f) elephant trunk, (g) echinoid, (h) Illex illecebrosus, (i) inchworm, and (j) snail feet. [123]
vering through water by actuating its tail. One of the advantages of a application in
water is that gravitational effects on the soft material are very low as the density of the
silicon rubber is similar to water. In [104, 106] (robots in Figures 3.2b and 3.2e) the
developed robot makes use of this property by conducting the experiments in a water
tank. This environmental advantage is even stronger in applications in space due to
the complete absence of gravity and surrounding medium. [121] Other applications of
soft robots are mobile robots for geometrically complex environments (Figure 3.2d) or
gripper for sensitive objects (Figure 3.2c).

Many soft robots are inspired by an elephant trunk, an octopus arm or a snake. So a
to lot of research has been done on the kinematics and control of those kind of robots. To
obtain a kinematical model [20, 10] derived equations using the tendon tress. In those
cases material properties of the soft deforming body have to be well known. Another
approach uses the displacement of tendons. [128] Those models however do not consider
outside forces or gravity. Pneumatically/hydraulically actuated robots don’t really have
this resource. That is why [90] use a mapping based on sample data. To obtain a
more accurate model [44, 87, 88, 42] use a Finite-Element-Method based approach. The
downside of those FEM simulations is that they are time consuming and therefor not
suited for real time control. Many different types of sensors have been used in order to
implement a feedback control:

- a Hall Effect sensor [92]
Figure 3.2: Different robot designs in literature
• a piezoelectric deflection sensor [113]

• a soft-strain sensor which utilizes conductive fabric [33]

• fiber-optic sensors [111]

• a flex bend sensor [52]

Sensors like the Hall Effect sensor [92], the piezoelectric deflection sensor [113] or the flex bend sensor [52] cause a nonuniform bending behavior for 3D bending which limits their applications. [105, 85, 81, 116, 57, 90, 52, 102, 109, 19, 128, 96]
4 Experimental Setup

This section describes the experimental setup used. This includes the design, fabrication and control of the robot as well as the mathematical model.

4 Robot Design

The designed robot is a arm shaped soft continuum robot made of silicon rubber. Embedded in the robot are 3 carbon-black impregnated rubber cords [2]. The same sensors have been used in [90]. The robot is driven by 3 DC-motors that pull on 3 tendons. The output torque is increased with a gearbox. To prevent the tendons from cutting through the silicon body plastic tubes are used to cover the tendons and protect the rest of the robot. A schematic of the robot is shown in Figure 4.1 and a cross section in Figure 4.2.

Additionally to the stretch sensors and the Hall-effect sensors, electromagnetic tracking sensors are used. Those sensors track the their location in an changing electromagnetic field. One of those sensors is placed at the tip of the robot and one at the base for reference. Using this system the data of the other sensors is validated. Also those sensors are used for calibration of the stretch sensors.

4 Fabrication

The silicon rubber used for the robot is a platinum-catalyzed silicon rubber [117]. Table 4.1 shows the material properties of the compared materials. In this project Ecoflex
Figure 4.1: Schematic of the robot (blue = silicon rubber; black = stretch sensors; yellow = cables; grey = plastic tubes)

Figure 4.2: Robot cross section
<table>
<thead>
<tr>
<th>Property</th>
<th>00-10</th>
<th>00-20</th>
<th>00-30</th>
<th>00-35</th>
<th>00-50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>1070 kg/m³</td>
<td>1070 kg/m³</td>
<td>1070 kg/m³</td>
<td>1070 kg/m³</td>
<td>1070 kg/m³</td>
</tr>
<tr>
<td>Cure Time</td>
<td>4 h</td>
<td>4 h</td>
<td>4 h</td>
<td>5 min</td>
<td>3 h</td>
</tr>
<tr>
<td>Tensile Strength</td>
<td>827 kPa</td>
<td>1103 kPa</td>
<td>1379 kPa</td>
<td>1379 kPa</td>
<td>2172 kPa</td>
</tr>
<tr>
<td>100% Modulus</td>
<td>55 kPa</td>
<td>55 kPa</td>
<td>69 kPa</td>
<td>69 kPa</td>
<td>83 kPa</td>
</tr>
<tr>
<td>max. Elongation</td>
<td>900 %</td>
<td>845 %</td>
<td>900 %</td>
<td>900 %</td>
<td>980 %</td>
</tr>
<tr>
<td>Shrinkage</td>
<td>&lt;0.1 %</td>
<td>&lt;0.1 %</td>
<td>&lt;0.1 %</td>
<td>&lt;0.1 %</td>
<td>&lt;0.1 %</td>
</tr>
</tbody>
</table>

Table 4.1: Material properties of Ecoflex[117]

00-30 was used as it is soft enough to be bend using small motors. Ecoflex 00-35 was rejected due to the low curing time which would not leave enough time to prepare the mixture and pour it into the mold.

After mixing the two components the mixture is placed in a vacuum chamber to extract air enclosed in it. The mixture is then removed from the chamber and poured into a mold (see Figure 4.3) and cured for at least 4 h. The curing does not take place inside the vacuum chamber as better results are obtained. Steel wires covered with plastic tubes serve as placeholders for the cables and sensors during the curing process. The placeholder for the sensors are removed from the cured arm, while the others remain as protection for the body from the tendons.

The tendons are fixed to the tip of the robot and at the base they are fixed to the DC-motors. When fixing the stretch sensors to the ends of the robot, the arm is first compressed by pulling the tendons. That way the stretch sensors remain under tension during the whole experiments, when the tendons are released. This pretension is necessary to prevent the sensors from rambling when the arm is compressed and thus improve
the quality of the readings.

During the whole fabrication lubrication is avoided as the sensors are very sensitive to silicon based environments. A more detailed characterization of this behavior is given in Section 5.

4 Control

For the control of the robot a Arduino Uno board was used. Figure 4.4 shows the electronic circuit.

The resistance of the stretch sensors are measured by measuring the voltage drop through the pull-down resistors $R_{pd} = 2 \, k\Omega$. For a known input voltage $V_{in} = 5 \, V$ The
Figure 4.4: Setup of the electronics (black = stretch sensors; green chips = Hall effect sensors)
resistance can then be calculated using Equation (4.1) (according to Section 2.2.1).

\[ R_{\text{sensor}} = R_{\text{pd}} \left( \frac{V_{in}}{V_{mes}} - 1 \right) \]  

(4.1)

The DC-motors are controlled by using one PWM output of the MCU and one digital output. By setting the voltage outputs of those pins according to Algorithm 1 a bidirectional actuation can be achieved.

<table>
<thead>
<tr>
<th>Algorithm 1: speed output decision logic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> speed</td>
</tr>
<tr>
<td>1 map speed to voltage $V_{out}$;</td>
</tr>
<tr>
<td>2 if $0 \leq V_{out}$ then</td>
</tr>
<tr>
<td>3    set PWM output to $V_{out}$;</td>
</tr>
<tr>
<td>4    set digital output to 0 V;</td>
</tr>
<tr>
<td>5    else</td>
</tr>
<tr>
<td>6    set PWM output to $5 , V + V_{out}$;</td>
</tr>
<tr>
<td>7    set digital output to 5 V;</td>
</tr>
<tr>
<td>8    end</td>
</tr>
</tbody>
</table>

To use the Hall-effect sensors as rotary encoders for the motors a permanent magnet is attached to the shaft of the motor (before the gearbox). Whenever the magnetic field aligns with the Hall-effect sensor a voltage peak (depending on power supply) at the output can be measured. This peak triggers a function in the MCU to increment or decrement the revolution counter.

The control commands for the robot are send from a computer running. So a serial connection via the USB port is established. Either the motor speed or a target position (target revolution counter) can be controlled. The control from the computer is done
by sending messages with a specific identifier character via serial port. The following identifiers have been implemented:

4 Kinematic

The mathematical description of the robot can be split into 4 spaces. Figure 4.6 illustrates those spaces.

The configuration space describes the kinematic of the robot through the homogeneous transformation matrix. Using a constant curvature model this matrix can be defined by the mean arm length $\ell = r\theta$, the curvature $\kappa = \frac{1}{r}$ and the bending angle $\phi$. The
<table>
<thead>
<tr>
<th>ID</th>
<th>Message example</th>
<th>Response example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>$r$</td>
<td>$R1234.5,987.0,1000.0,500,200,1234$</td>
<td>This requests a sensor reading. The response consists of the 3 stretch sensors’ resistances and the revolution counters.</td>
</tr>
<tr>
<td>s</td>
<td>$s.9999,.0400,-.321$</td>
<td></td>
<td>This sets the motor speeds to the 3 values (mapped from $-1$ to $1$ to $-5$ V to $5$ V).</td>
</tr>
<tr>
<td>t</td>
<td>$t100,200,0$</td>
<td></td>
<td>This sets the target counter to the 3 values and starts moving the motors towards them.</td>
</tr>
<tr>
<td>c</td>
<td>$c2000$</td>
<td></td>
<td>This calibrates the Hall-sensors by pulling on every cable with the maximum force and then setting the counter value to the input value.</td>
</tr>
<tr>
<td>a</td>
<td>$a900$</td>
<td></td>
<td>This sets all counter values to the input value (used for manual adjustments).</td>
</tr>
</tbody>
</table>

Table 4.2: Commands for control via serial connection with the MCU
Rotation with the angle $\phi$ around the $z$-axis can be described with Equation (4.2). The transformation for the bending (in the $xy$-plane) is described by Equation (4.3). The transformation for the robot can then be written as product of all transformations (see Equation (4.4)). [128]

$$T_{\phi} = \begin{bmatrix}
\cos \phi & \sin \phi & 0 & 0 \\
-\sin \phi & \cos \phi & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$ \hspace{1cm} \text{(4.2)}$$

$$T_{r,\theta} = \begin{bmatrix}
\cos \kappa \ell & 0 & -\sin \kappa \ell & r (1 - \cos \kappa \ell) \\
0 & 1 & 0 & 0 \\
\sin \kappa \ell & 0 & \cos \kappa \ell & r \sin \kappa \ell \\
0 & 0 & 0 & 1
\end{bmatrix}$$ \hspace{1cm} \text{(4.3)}$$
The task space is the space in which the end-effector moves. It is defined by 3 spatial coordinates \((x, y, z)\). Assuming the motors are not operating at their force maximum, the tendon lengths lie in a constant range. The task space for this case is visualized in Figure 4.7.

For the mapping from the configuration space to the task space \(f_{c,x}\) only the last
column of the transformation matrix is required (see Equations (4.5) to (4.7)). [128]

\[
x = \frac{\cos \phi (1 - \cos \kappa \ell)}{\kappa}
\] (4.5)

\[
y = \frac{\sin \phi (\cos \kappa \ell - 1)}{\kappa}
\] (4.6)

\[
z = \frac{\sin \kappa \ell}{\kappa}
\] (4.7)

For the transformation from the task space to the configuration space \( f_{x,c} \) those equations can be solved for the configuration variables (see Equations (4.8) to (4.10)). [128]

\[
\kappa = \frac{2\sqrt{x^2 + y^2}}{x^2 + y^2 + z^2}
\] (4.8)

\[
\phi = \text{atan2} (y, x)
\] (4.9)

\[
\ell = \frac{1}{\kappa} \arcsin (z\kappa)
\] (4.10)

The actuator space \( \mathbf{q} \) consists of the 3 cable lengths \((l_1, l_2, l_3)\). As those actuators are linear independent, a closed form mapping from actuator space to configuration space \( f_{s,c} \) exists. With \( \varphi_1 = \frac{1}{2}\pi, \varphi_2 = \frac{7}{6}\pi \) and \( \varphi_3 = \frac{11}{6}\pi \) those equations are shown in
Equations (4.11) to (4.13)). \[128\]

$$\kappa = 2\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_1l_3 - l_2l_3} \over \delta (l_1 + l_2 + l_3) \quad (4.11)$$

$$\phi = \text{atan2} \left( l_2 + l_3 - 2l_1, \sqrt{3} (l_2 - l_3) \right) \quad (4.12)$$

$$\ell = \frac{l_1 + l_2 + l_3}{3} \quad (4.13)$$

The mapping from configuration space to actuator space $f_{c,q}$ can be described with Equation (4.14). \[128\]

$$l_i = \ell (1 - \kappa \delta \cos (\varphi_i - \phi)) \quad (4.14)$$

The sensor space consists of the 3 sensor lengths ($s_1, s_2, s_3$). As the sensor are rotated by 30° to the cables, the equations for the mapping between sensor space and configuration space ($f_{s,c}$ and $f_{c,s}$, see Equations (4.15) to (4.18)) look similar to the mapping between actuator space and configuration space. \[128\]

$$\kappa = 2\sqrt{s_1^2 + s_2^2 + s_3^2 - s_1s_2 - s_1s_3 - s_2s_3} \over \delta (s_1 + s_2 + s_3) \quad (4.15)$$
\[ \phi = \text{atan2} \left( s_2 + s_3 - 2s_1, \sqrt{3} (s_2 - s_3) \right) - \frac{\pi}{6} \]  
(4.16)

\[ \ell = \frac{s_1 + s_2 + s_3}{3} \]  
(4.17)

\[ s_i = \ell \left( 1 - \kappa \delta \cos (\alpha_i - \phi) \right) \]  
(4.18)

A more general mathematical model that does not assume the cables and sensors to be evenly distributed is shown in Appendix A.
5 Material Characterization

[33] used a conductive textile (Electrolycra, from Mindsets Ltd, United Kingdom) which is efficient for strain levels up to 30%. For the mentioned conductive textile curves, the resistance does not change beyond 30% of the engineering strain. On the other hand, rubber bands can work effectively for 10 unitcommands times higher strain levels (up to 300% strain levels).

For using the Ecoflex 00-30 elastomer as a body of the soft robotic arm and also intensively studying the rubber band, the following things are really important for us. Tensile test (stress-strain constitutive relation), Resistance-tension (Resistance-Strain curve) Cyclic tensile test (Mechanical Hysteresis behavior), Resistance cyclic test (Resistance cyclic test).

Material testing has been done following the instruction from ASTM D412 - 16) Standard Test Methods for Vulcanized Rubber and Thermoplastic Elastomers—Tension(. First, using 3d printing and rapid prototyping techniques the mold for the tensile specimens has been made. The specimens made by degassing the liquid mixture of the elastomer in vacuum, following up with a 2 h to 3 h of curing time. Mullin effects has been removed by stretching the specimen for several times. The dimension for elastomer specimen is 4.2 mm × 12 mm × 35.88 mm. The rate of the experiment (i.e. the strain rate) was set to be 60 mm/min. We also tried higher speed (e.g. 200 mm/min) and we did not observe significant strain rate sensitivity, which is also in agreement with the
Figure 5.1: Cyclic stress-strain curve for the Ecoflex 00-30 which shows negligible hysteresis.
Figure 5.2: Cyclic stress-strain curve for the conductive rubber cord [2] which shows significant hysteresis effect.

literature, as reported by the [88]. They reported that there is less than 3% viscoelastic effect for Eco-flex 00-30. The test stand used for the material testing experiment is Mark 10, ESM 303 (C.S.C Force Measurement, Inc.).

Figure 5.2 shows the cyclic stress-strain loading curve for the rubber band for 20 cycles. We observe a significant hysteresis effect on this curve. Therefore we observe a large phase difference between the stress and the strain. Here if we denote the area under the loading curve (upper curves) by and denote the area for the unloading curve (lower curves) by . We can compute the dissipative energy compare to the unloading curve which is equal to 59.99%. In addition to hysteresis effect we can also observe the stress/force relaxation effect. As we notice a band width for the upper curve and the
Figure 5.3: Steady Resistance- strain curve for the conductive rubber band [2] up to 20% engineering strain, which shows significant aging effect.

lower curve in Fig2. Meaning, during the experiment by repeating the cycles the plot has a shift to the below.

In Figure 5.3 we show the experimental result for the aging effect of the rubber band. By aging here we mean the life of the rubber band inside the elastomeric arm. We noticed a different behaviour for the aged rubber bands. Due to diffusivity of the silicone rubber particles into the rubber band the rubber resistance will increase on a timely manner. The blue curve (top curve) is the rubber band resistance after 2 days, which shows two to three order of magnitude increase in resistant. However, it also changes the behavior of the material for the strains beyond 17% and we do not observe the twist anymore. After running several experiments, the data from the aged rubber band has been more
useful in terms of calibrating the sensors. Further research should account for the tuning and regulating the rubber bands resistance and its sensitivity to the environment.

For measuring the real-time resistance of the conductive rubber, an Arduino UNO microcontroller was utilized and programmed using MATLAB Arduino package. Wires for measuring the real-time resistance were attached close to the edge of the tensile stand’s grippers. Resistance data acquisition has been synchronized with the displacement data from the tensile experiment.
6 Robot Control

This section describes the robot control. First a calibration process for the robot is presented. In the next steps the robot is tested in an open-loop and an closed-loop environment.

6 Calibration

Because the behavior of the conductive rubber bands doesn’t follow a linear pattern the robot is calibrated before every experiment. During this calibration process multiple data points are collected with their respective length (obtained from EM-senors). For those points a polynomial of 3rd order is fitted and used to convert from sensor reading to sensor length. The calibration algorithm is shown in Algorithm 2. Figure 6.1 shows the sensor readings and the sensor length during the calibration process. The data shows that the resistance of the peaks decreases over time. The biggest change of the resistance range happens during the first cycle. To get more consistent data for the calibration, the first cycle is cut off and thereby not used for the calibration. This cycle shows significantly different results due to the Mullins effect. Figure 6.2 shows the polynomial regression for those data points. Even though all sensors function in a different resistance range, the basic shape of the data matches.
Algorithm 2: calibration process

Input: max values $c_{\text{max}}$

Input: number of steps $N_{\text{steps}}$

Input: number of cycles $N_{\text{cycles}}$

Result: polynomial regression function for each sensor $s_i(R_{si})$

1. for $n_c \leftarrow 0$ to $N_{\text{cycles}}$ by 1 do
2.     for $c \leftarrow c_{\text{max}}$ to 0 by $\frac{1}{N_{\text{steps}}}$ do
3.         move motors to counter position $c$;
4.         read sensor resistances $R_{si}$;
5.         read robot length using EM sensor $\ell_i$;
6.     end
7.     for $c \leftarrow 0$ to $c_{\text{max}}$ by $\frac{1}{N_{\text{steps}}}$ do
8.         move motors to counter position $c$;
9.         read sensor resistances $R_{si}$;
10.    read robot length using EM sensor $\ell_i$;
11. end
12. end
13. calculate polynomial regression for data of each sensor $s_i(R_{si})$;

Figure 6.1: Calibration process
6 Open-Loop Control

To test the open-loop behavior of the robot a circular trajectory was discretized at to a number of points. First the robot undergoes a calibration step to set the initial state to a known position. During this calibration the motors pull on every cable with the equal maximum force and after that release all cables equally. The end-effector is then moved to the predefined trajectory points by pulling and releasing the cables to the required lengths (calculated with the inverse kinematic).

Figure 6.3 shows the robot at different points along the experiment. The end-effector coordinates acquired from the Hall effect sensors, EM sensor and the stretch sensors are shown in Figure 6.4. Figure 6.5 presents the same data over time.

The Figures 6.4 and 6.5 show that the basic shape of the trajectory can be recon-
Figure 6.3: Open-loop recording

Figure 6.4: Open-loop end-effector coordinates
Figure 6.5: Open-loop end-effector coordinates over time
structed. However the data in Figure 6.5 shows that with increasing time the amplitude of the stretch sensors decreases in the same way as observed during the tensile testing (Section 5). Despite this fact the direction toward which the arm is bending still matches the direction obtained from the stretch sensors.

Also it can be observed that the x coordinate of the trajectory does not reach the expected position. The reason for that may be friction forces between the cable located at \((-r_{\text{cable}}, 0)^T\) and robot. Even though the motor releases the cable by the right amount the friction prevents the arm to reach the desired position. Using lubrication could reduce the friction. However as the stretch sensors are very sensitive to silicon based liquids, lubrication the arm was avoided.

6 Closed-Loop Control

In this section the stretch sensors are used in a closed-loop control to actuate the robot to a defined target. In this experiment a PI-controller was used. A low pass filter (moving average filter) was implemented to remove high frequency noise of the sensor readings. The controller setup is shown in Figure 6.6. Here \(r(t)\) is the target cable length, \(u(t)\) is the speed input for the DC motor and \(y(t)\) is the cable length calculated from the sensor reading. For simplification only one motor was actuated.

Figure 6.7 shows some frames of the experiment.

Figure 6.8 shows the results of this experiment. The data obtained from the stretch sensors can be used to control the robot to specified position. However the viscoelastic
Figure 6.6: PI controller

Figure 6.7: Closed-loop recording
properties of those sensors make it difficult to actuate the robot fast, as in those cases the sensors become inaccurate or require taking the viscoelastic behavior into account.
7 Conclusion and Outlook

Soft continuum robotic arms, which combine the dexterity and high degree of freedom of continuum robots, are able to escape many of the pitfalls of their rigid and hard continuum counterparts due to their use of hyperelastic materials [110], and better mimic the gentleness, articulation, and high level of control and haptic ability found in nature. Due to the low-stiffness and light-weight materials, much of the power demand seen with conventional rigid or continuum hard robots is unnecessary, allowing for the use of batteries and small DC motors to provide the motive force for actuation. It is because of this that soft continuum robots prove to be exceptional candidates for modular design, whereby each module contains its own source of actuation and control, making it possible to design extendable and/or more highly complex soft robotic arms with unique functionalities.

Using carbon impregnated rubber cords as stretch sensors to sense the current state of a soft robot presents a lot potential as material properties of the soft robot are not required for the kinematic model. However those stretch sensors bring many challenges along. It is hard to maintain consistent results over longer periods of time due to high environmental dependencies of the sensors, which makes calibration before usage unavoidable. That aside even within a single experiment consistency is not guaranteed. A more accurate model of the stretch sensors including their hysteresis effect would be needed.

As the created robot uses small actuators and electronic parts this design is suited to
be extended to a multi-section robot. As the robot would be able to carry the weight of its hardware, those parts could be combined in a small module that is connected wireless (over an HTTP server) to a computer that gives control commands. A robot like this could have many different applications as it is easy extendable by adding more sections.
A General Mathematical Model

Forward Kinematic

For the kinematic a constant curvature model is assumed. The length of a thread along the robot is therefore only dependent on the distance to the bending axis. The function \( L(X,Y) \) describes this length for any thread. The equipotential curves (curves of constant value) are lines parallel to the bending axis.

The length \( L(R) \) of a thread with distance \( R \) to the bending axis can be calculated using Equation (A.1).

\[
L(R) = R\theta
\]  

(A.1)

Because the length \( L \) is linear proportional to the distance \( R \), the function \( L(X,Y) \) has to be a plane.

\[
L(X,Y) = a_1 + a_2X + a_3Y
\]  

(A.2)

With the known sensor length the coefficients \( a_i \) can be calculated by solving the following linear system of equations:

\[
\begin{bmatrix}
1 & X_{s1} & Y_{s1} \\
1 & X_{s2} & Y_{s2} \\
1 & X_{s3} & Y_{s3}
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
a_3
\end{bmatrix}
= 
\begin{bmatrix}
s_1 \\
s_2 \\
s_3
\end{bmatrix}
\]  

(A.3)

The gradient of the function \( \nabla L(X,Y) \) gives the direction \( \vec{n} \) toward which the arm is
bending.

\[
\nabla L (X,Y) = \begin{bmatrix} a_2 \\ a_3 \end{bmatrix}
\]

(A.4)

\[\|\nabla L (X,Y)\| = \vec{n}\]

(A.5)

For \(\vec{n} \neq \vec{0}\) the arm is bend. To determine the bending angle \(\phi\) and the bending radius \(R(\vec{0}) = r\) of a line going through the center of the arm, two length values \(L(\vec{0})\) and \(L(\vec{n})\) are used. With Equation (A.6) the values \(\phi\) and \(r\) can be calculated.

\[\phi = \frac{L(\vec{0})}{R} = \frac{L(\vec{n})}{R(\vec{n})}\]

(A.6)

\[r = \frac{L(\vec{n})}{L(\vec{0}) - L(\vec{n})}\]

(A.7)

\[\phi = \frac{L(\vec{0}) \left( L(\vec{0}) - L(\vec{n}) \right)}{L(\vec{n})}\]

(A.8)

Assuming the arm only bends in the x-z-plane, the end-effector position can be calcu-
lated with the homogeneous transformation in Equation A.9

\[
    T_{r, \theta} = \begin{bmatrix}
        \cos \theta & 0 & -\sin \theta & r \left( 1 - \cos \theta \right) \\
        0 & 1 & 0 & 0 \\
        \sin \theta & 0 & \cos \theta & r \sin \theta \\
        0 & 0 & 0 & 1
    \end{bmatrix}
\]  

(A.9)

To get the actual end-effector position a rotation to the bending direction has to be added:

\[
    \phi = \text{atan2} \left( n_y, n_x \right)
\]

(A.10)

\[
    T_{\phi} = \begin{bmatrix}
        \cos \phi & \sin \phi & 0 & 0 \\
        -\sin \phi & \cos \phi & 0 & 0 \\
        0 & 0 & 1 & 0 \\
        0 & 0 & 0 & 1
    \end{bmatrix}
\]  

(A.11)

The combined transformation matrix from the base to the end-effector is then dependent on \( \phi, \theta \) and \( r \):

\[
    T(\phi, \theta, R) = T_{\phi} T_{r, \theta}
\]

(A.12)
Inverse Kinematic

For the inverse kinematics the position of the end effector $\vec{p}$ is known and the cable length is determined. In the first step the direction in which the arm bends is calculated. This is done by projecting the vector $\begin{bmatrix} x & y & z \end{bmatrix}^T$ onto the xy plane. Equation (A.13) shows the bending direction vector in a normalized form.

$$
\vec{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \frac{1}{\sqrt{x^2 + y^2}} \begin{bmatrix} x \\ y \end{bmatrix}
$$

(A.13)

The end effector position is then rotated to the x-z-plane:

$$
\vec{p} = T_2^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix}^T
$$

(A.14)

Using the translational part of Equation (A.9) the radius $r$ and the angle $\theta$ can be calculated (see Equations (A.15) and (A.16)).

$$
(p_x - r)^2 + p_z^2 = r^2 (\cos^2 \theta + \sin^2 \theta) = r^2 \Rightarrow r = \frac{p_x^2 + p_z^2}{2p_x} \quad \text{(A.15)}
$$

$$
p_z = r \sin \theta \Rightarrow \theta = \arcsin \frac{2p_x p_z}{p_x^2 + p_z^2} \quad \text{(A.16)}
$$

To obtain the cable length the radii for the cables $r_{ci}$ are required. The radii can be
calculated by projecting the coordinates of the cables \( \begin{bmatrix} x_{ci} \\ y_{ci} \end{bmatrix}^\top \) (x and y at the base of the arm) onto the bending direction vector \( \vec{n} \). The length of the component along \( \vec{n} \) is the subtracted from \( r \) to obtain the radius for a cable \( r_{ci} \) (see Equation (A.17)).

\[
    r_{ci} = r - \frac{\begin{bmatrix} x_{ci} \\ y_{ci} \end{bmatrix} \cdot \vec{n}}{\| \vec{n} \|^2}
\]  

(A.17)

Using Equation (A.18) the length for an cable can then be calculated.

\[
    L(x_{ci}, y_{ci}) = r_{ci} \theta
\]  

(A.18)
B Code Instructions

The code that is flashed on the MCU expects a hardware setup as shown in Figure 4.4. If changes are made the pin numbers have to be updated in the Arduino code ("Arm.cpp" and "main.ino").

Flowcharts for the matlab codes are given in Figures B.1 to B.5. Other files may be required but are self explanatory such as “rotx.m” which returns a rotation matrix around the x-Axis.
Figure B.1: experiment steps
send read request by writing 'r' to serial port

answer did not arrive

try reading answer True

reading False

10 unsuccessful attempts False

resend request True

Figure B.2: flow chart for “readSensors.m”
pull on every cable with maximum force
wait until static state is reached
release cables by the same amount (counter)
manual adjustment is required

adjust manually (code in file as comment)

Figure B.3: flow chart for “calibration.m”
Figure B.4: flow chart for “runThroughPts.m”
target not in threshold

read sensors

readings

resend target with "sendValues.m"

send target with "sendValues.m"

Figure B.5: flow chart for “move2Pos.m”
Bibliography


[44] Yahya Elsayed, Augusto Vincensi, Constantina Lekakou, Tao Geng, C. M. Saaj, Tommaso Ranzani, Matteo Cianchetti, and Arianna Menciassi. Finite Element


68

[57] Michael W. Hannan and Ian D. Walker. Kinematics and the Implementation of  
an Elephant’s Trunk Manipulator and Other Continuum Style Robots. Journal of  


[59] Shigeo Hirose. Biologically inspired robots: snake-like locomotors and manipulators,  

[60] Filip Ilievski, Aaron D Mazzeo, Robert F Shepherd, Xin Chen, and George M  
2011.

[61] Filip Ilievski, Aaron D. Mazzeo, Robert F. Shepherd, Xin Chen, and George M.  
Whitesides. Soft robotics for chemists. Angewandte Chemie - International Edi-


Bio-inspired crawling locomotion of a multi-arm octopus-like continuum system.  
IEEE International Conference on Intelligent Robots and Systems, pages 145–150,  
2012.


[89] NDI. Aurora tracking sensors.


[97] Ryan S Penning, Jinwoo Jung, Justin Borgstadt, Nicola J Ferrier, Michael R Zinn, and Others. Towards closed loop control of a continuum robotic manipulator for


[126] Stephen Tully, George Kantor, Marco a Zenati, and Howie Choset. Shape Estima-
tion for Image-Guided Surgery with a Highly Articulated Snake Robot Highly


[128] Robert J. Webster and Bryan A. Jones. Design and kinematic modeling of constant
curvature continuum robots: A review. International Journal of Robotics Research,

[129] Michael Wehner, Ryan L. Truby, Daniel J. Fitzgerald, Bobak Mosadegh, George M.
Whitesides, Jennifer A. Lewis, and Robert J. Wood. An integrated design and

[130] Kai Xu and Nabil Simaan. An investigation of the intrinsic force sensing capabilities

[131] Y. Yekutieli. Dynamic Model of the Octopus Arm. II. Control of Reaching Mov-

[132] Michael C. Yip and David B. Camarillo. Model-Less Hybrid Position/Force Con-
trol: A Minimalist Approach for Continuum Manipulators in Unknown, Con-