University of Twente

EEMCS / Electrical Engineering
Robotics and *Mechatronics*

Image-Guided Control of a Continuum Robot

S.L. (Sander) Janssen

MSc Report

Committee:
Prof.dr.ir. S. Stramigioli
Dr. S. Misra
J. Jahya, MSc
R.J. Roesthuis, MSc
Ir. E.E.G. Hekman

November 2012

Report nr. 030RAM2012
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

This study presents the development of a continuum robot which could serve as part of a surgical robot. Continuum robots, because of their dexterous nature and continuous shape are suited for minimally invasive surgery. Their inherent flexibility also reduces the risk of damage. Continuum robots have these properties due to the absence of discrete joints. Previous research in the domain of continuum robots has focused on developing kinematic and dynamic models. These models attempt to take in account as much of the inherent imperfections such as friction and fabrication errors as possible. This study instead focussed on showing that a simple kinematic model based on the ideal behaviour of the robot is sufficient when combined with a closed-loop controller with 3D position feedback.

A continuum robot and its actuator have been developed. The continuum robot was tendon actuated. It consisted of a flexible nitinol backbone with four disc shaped tendon guides fixed equidistantly to each other to it. Two designs of robot were created: A 160 mm length robot with 5 mm radius tendon guides and a 160 mm one with 10 mm radius tendon guides. The actuator was fitted with four motors to pull the tendons and had a translation stage for an insertion movement.

A simple kinematic model based on the constant curvature principle was developed. This principle assumes that the backbone bends in a circular arc. A PID-controller was implemented. This PID-controller was modified to improve performance with non-continuous input signals. A stereoscopic camera setup provided the feedback for this controller. An image processing algorithm took the images provided by the cameras and triangulated the 3D position of the last tendon guide. The controller, kinematic model and image processing were implemented as a multi-threaded C++ win32 application. The complete system worked at 15 Hz.

Experiments were performed with the 20 mm radius design to compare the performance of the open-loop system with the closed-loop system. Every experiment was performed 5 times. While the open-loop system had a 60 mm step response steady state error of approximately 15 mm, the closed-loop system had a steady state error of about 2 mm. 3D shapes were used as input as well. The most complex of these shapes was a series of squares that were sequentially shifted in space, forming a beam shape. The open-loop system had an average tracking error of 20 mm while tracking this shape. The closed loop system had an average tracking error of 6 mm. A 3D haptic device was also used as an input for the system. The system was capable of tracking the input of a human following a cross shape pattern with an average tracking error of approximately 4 mm. The 5 mm radius robot was also tried, but due to excessive friction the robot could not be controlled in a stable manner.

This study showed that a simple kinematic model combined with a 3D position feedback controller is an alternative for a more complex kinematic model. Future research involves embedding fibre Bragg grating strain sensors in the backbone of the robot. This would allow the reconstruction of the robot shape in 3D while *in vivo*.

Contents

Chapter 1

Introduction

Minimally invasive surgery has been a major breakthrough in surgery. It reduces scarring and the recovery time of the patient. A type of minimally invasive surgery is natural orifice surgery (NOS). In this type of surgery the surgical instrument is inserted through an already existing opening in the body. An example of NOS is natural orifice translumenal endoscopic surgery (NOTES) [21]. In this case, an abdominal operation is performed where the surgical instrument is inserted through a natural orifice, such as the mouth.

Robots can play an important role in performing minimally invasive surgery. The da Vinci Surgical robot of Intuitive Surgical [18], as can be seen in figure 1.1, is an example of a commercially available robot that can be used to perform minimally invasive surgeries. However, this machine is directly controlled by the surgeon and is not an autonomous device. While there are advantages to the surgeon being totally in control, humans are still prone to mistakes. Therefore it is desirable to have a robot that can perform operations on its own or assist the surgeon to improve his efficacy during surgery. This robot has to be precise so that surgeries can be performed with as little trauma to the patient as possible.

Figure 1.1: The da Vinci surgical robot

1.1 Continuum Robots

This study presents the development of a continuum robot which could serve as part of a surgical robot. A continuum robot can be defined as a continuously bending robot

with an elastic structure [19]. Continuum robots, because of their dexterous nature and and continuous shape due to the absence of discrete joints, are very suited for minimally invasive surgery. Their inherent flexibility also reduces the risk of damage. They are relatively light and can be controlled to perform precise movements [12]. There are several different kind of continuum robots: Tendon actuated robots [5, 20, 2], pneumatic muscle actuated robots[13] and others. Two examples of totally different continuum robots are shown in figure 1.2. Continuum robots have been the subject of intensive research for several years now [19, 16].

(a) Pneumatic tendon robot (b) A concentric tube robot

Figure 1.2: Types of continuum robot

Research has focussed on the kinematic and dynamic modelling of continuum robots [10]. These models are capable of describing the behaviour of the robots. Many of the kinematic models are based on the constant curvature principle. This principle assumes that when a constant moment is applied along a beam, the beam bends in a circular arc. Models using this assumption have been used to control continuum robots [10, 6, 5]. However, Some of the effects that happen inside the continuum robot such as friction and fabrication imperfections are difficult to accurately model. This is why more complex models are being developed that account for these effects. Some of these complex kinematic and dynamical models are promising, but none of these models have been verified in an experimental setup [4].

The performance of these kinematic models robots can also be improved through feedback control [1]. This has already been done by tracking of the tip position in 2D [14, 9] with a camera. This study expanded on this concept by using an optical 3D tracking system. While doing optical detection *in vivo* is impossible. This study was a proof of concept and the optical system could be replaced in future research by another imaging modality. For example, ultrasound has been proven to be capable of tracking objects like continuum robots *in vivo* [15].

1.2 Assignment

The focus of this assignment was not on developing a complex kinematic model of the continuum robot which captures all the effects as most research focusses on. This study focussed on showing the concept of driving the robot using a simple kinematic model and then combining it with a 3D position feedback controller to improve its performance in real-time.

To make this possible several steps needed to be performed.

- Design a continuum robot

- Design an actuator
- Design a controller
- Create a feedback system
- Implement/fabricate the designs
- Evaluate the performance

The kinematic model based on the constant curvature assumption is derived in chapter 2. Continuum robots come in different shapes and sizes. In this case a tendon actuated robot was chosen. This design was chosen because it was simple to manufacture and actuate. It is described in section 3.2. To actuate the robot an actuation device was designed and built. This actuator is discussed in section 3.3. A controller was designed and implemented, it is described in chapter 4. This study was a proof of concept and therefore an easy to implement feedback system was chosen: an optical system using CCD cameras. The software that ran the system, its structure and functionality is discussed in chapter 5. Experiments were performed to quantify the performance. These experiments are discussed in chapter 6. The assignment was reviewed and recommendations were given, these conclusions are written down in chapter 7.

1.3 Contributions

During this assignment some contributions were made.

- Designed a continuum robot
- Created an actuator capable of driving different designs of continuum robot
- Created a thread-based software framework for closed-loop control of a continuum robot
- Showed that 3D position feedback control of a continuum robot is possible

In the next chapter the kinematic model will be discussed.

Chapter 2

Kinematic model

This chapter discusses the kinematics of a continuum robot with a flexible backbone. The reason why this design was chosen is explained in section 3.2. The robot as shown in figure 2.1 was designed in this study. This design had four tendons set at 90° angles from each other. This configuration allowed the robot to bend in all directions. The four tendon flexible backbone layout was assumed in this model. Other variables such as device length, number of tendon guides, tendon guide radius etc. can be varied in this model.

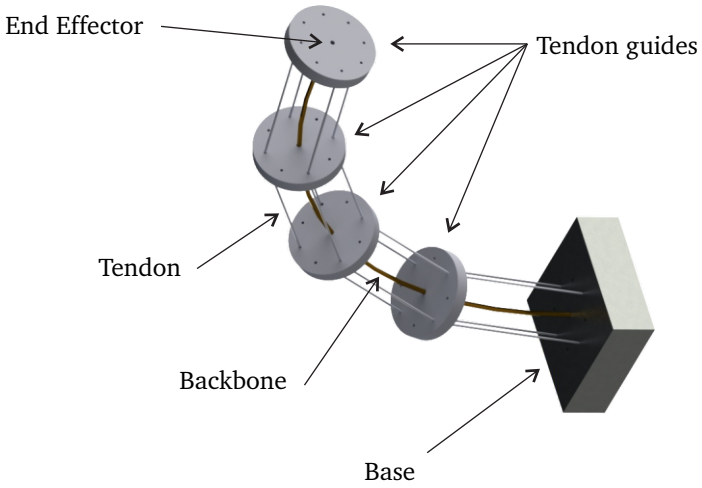


Figure 2.1: The tendon actuated continuum robot

2.1 Goal

The purpose of this kinematic model was to derive the equations how the tendon lengths affect the end-effector. The end-effector of the continuum robot is defined as the center of the last tendon guide. The equations that show the relations from tendon lengths to end-effector position (X_e, Y_e, Z_e) are derived in section 2.2. To drive the continuum robot the inverse kinematics need to be known. These are discussed in section 2.3.

The model discussed in this chapter assumed that when the robot was actuated the backbone bent in a perfect circular arc, this is the so-called constant curvature model. The constant curvature model has proven a capable method of describing the behaviour of this type of continuum robots [10, 6, 5]. But it has limitations. This is due to the

fact that an actual continuum robot differs from the ideal case. This may be due to fabrication imperfections such as misalignment of the tendon guides. The model also does not account for effects such as stretching of the tendons or friction. However, the focus of this study was to show that a simple kinematic model could be combined with a feedback system. Therefore, this model was suitable for this study. All the variables used in this model are listed in table 2.1.

Variable	Description	Figure
A	The deflection in the plane of bending	2.5
d	Distance from backbone to tendon guide hole	2.4
i	Number of the tendon	2.3b
ϕ	Angle at which the robot is bending in the (X, Y) plane	2.3b
ϕ_i	Angle of the ith tendon hole in relation to ϕ	2.3b
l_b	Length of the backbone of the continuum robot	2.3a
l_c	Centre point distance in a segment	2.4
l_i	Length of the ith tendon	2.4
l_{lin}	The position of the linear stage	
l_y	The length of the robot in the Z-direction	2.5
n	Number of segments of the robot	
R	Radius of the circular arc describe by the robot	2.3a
R_i	Radius of the circular arc in relation to the ith tendon hole	2.4
θ	Angle of the circular arc described by a segment of the robot	2.4
(X_e, Y_e, Z_e)	The position of the end-effector	2.3

Table 2.1: Description of the variables used in the kinematic model

2.2 Forward Kinematics

The forward kinematics describe the relations between the tendon lengths, the shape of the continuum robot and the position of the end effector. The origin of the coordinate system is defined as the point where the continuum robot leaves the insertion stage. This point is shown in figure 2.2. The actual design of the continuum robot and its actuator is discussed in chapter 3. The actuator consisted of a carriage to which the continuum robot is connect. This carriage had four motors attached to it to pull the tendons. This carriage was mounted on a linear stage for an insertion movement. This linear stage moved the continuum robot in the Z direction.

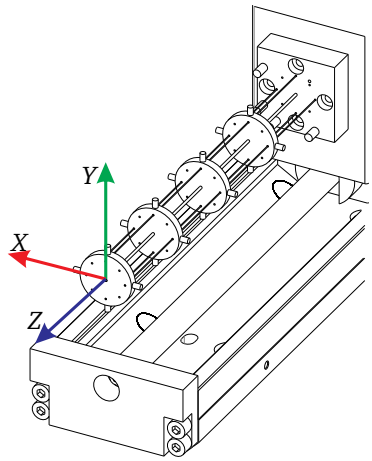
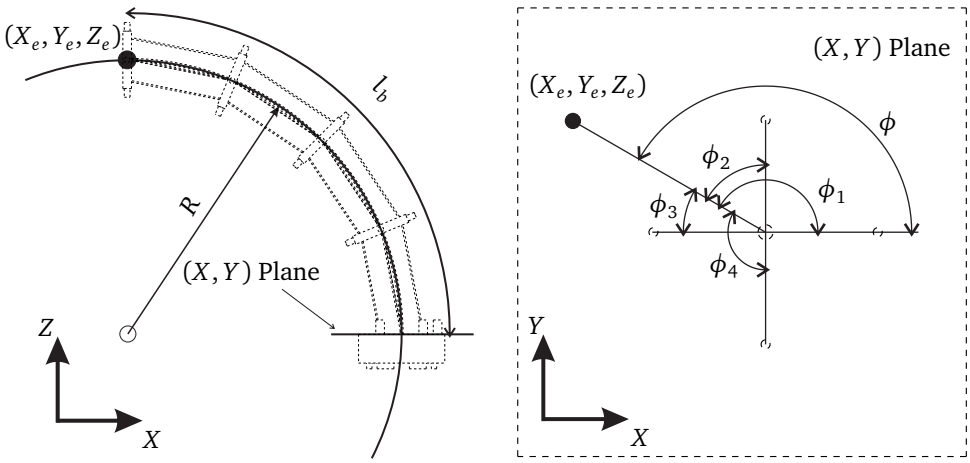


Figure 2.2: The origin of the system

A change in the tendon lengths (l_1, l_2, l_3, l_4) results in movement of the end-effector of the robot. This means that the robot bends in a circular arc of length l_b with radius R and at the angle ϕ. R is the radius of the circular arc which the backbone describes. l_b is the length of the backbone of the robot, which is the length of the circular arc. ϕ is the angle at which the backbone curves in the (X, Y) plane. The position of the end-effector is defined as (X_e, Y_e, Z_e). This configuration is shown in figure 2.3.



(a) The radius of curvature R (b) The different ϕs

Figure 2.3: Angle ϕ and radius of curvature R when the robot is bending

As can be seen in figure 2.3b per tendon a ϕ_i is defined, in this case i denotes the tendon number. This is the angle at which backbone is protruding from the base of the robot in relation to the tendon. The different ϕ_is are related to ϕ in the following manner:

$$\phi_1 = \phi, \phi_2 = \frac{1}{2}\pi - \phi, \phi_3 = \pi - \phi, \phi_4 = \frac{3}{2}\pi - \phi \tag{2.1}$$

It was assumed that the tendon lengths l_i were the same as the backbone length l_b when the continuum robot was straight. A segment of a continuum robot while bending is shown in figure 2.4. Be aware that this figure shows the device in the case where ϕ is equal to one of the ϕ_is, this is only the case when only one tendon is being actuated.

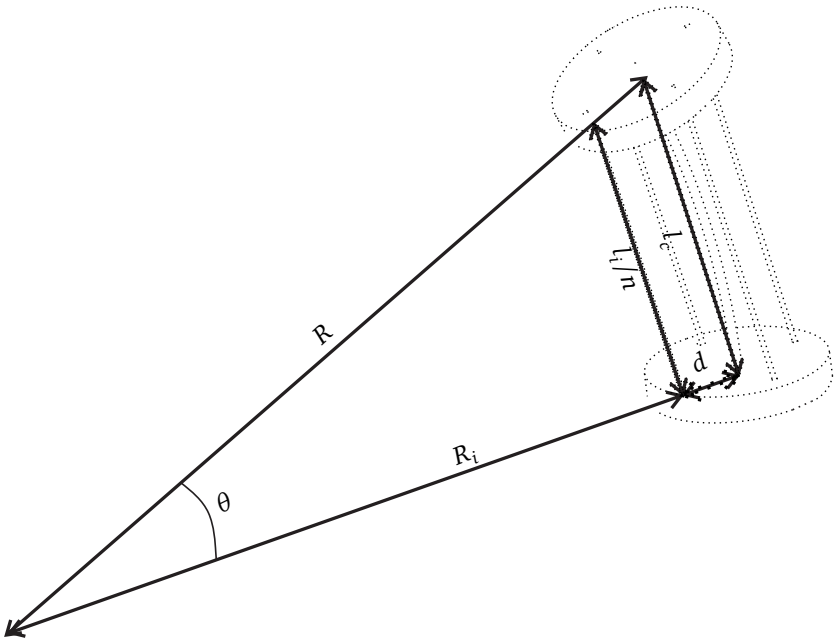


Figure 2.4: A segment of the continuum robot in a bent configuration

The tendons run in a straight line from tendon guide to tendon guide. In this model we assumed that the tendon guides are infinitely thin. We can define a straight line l_c which runs from backbone hole to backbone hole in the tendon guides, l_c. This line is the same length the average of the tendon lengths within the segment.

$$l_c = \frac{l_1 + l_2 + l_3 + l_4}{4n} \tag{2.2}$$

This line can be expressed in terms of θ and R.

$$l_c = 2 \cdot R \cdot \sin(\theta/2) \tag{2.3}$$

We can define another radius: R_i. This radius is from the center of the circle to the ith tendon hole. This radius can be defined by a geometric relation.

$$R_i = R - d \cdot \cos\phi_i \tag{2.4}$$

We can use (2.4) to get the tendon length inside the segment.

$$l_i/n = 2 \cdot R_i \cdot \sin(\theta/2) \tag{2.5}$$

If we combine (2.5) and (2.4) we get another expression for the tendon length inside the segment.

$$l_i/n = 2 \cdot (R - d \cdot \cos(\phi_i)) \cdot \sin(\theta/2) = \overbrace{2 \cdot R \cdot \sin(\theta/2)}^{l_c} - 2 \cdot d \cdot \cos(\phi_i) \cdot \sin(\theta/2) \tag{2.6}$$

We can use this equation to get an expression for l_c in terms of tendon length and θ.

$$l_c = l_i/n + 2 \cdot d \cdot \cos(\phi_i) \cdot \sin(\theta/2) \tag{2.7}$$

This gives us an relation between the different tendons, for example l_1 and l_3

$$l_1/n + 2 \cdot d \cdot \cos(\phi_1) \cdot \sin(\theta/2) = l_3/n + 2 \cdot d \cdot \cos(\phi_3) \cdot \sin(\theta/2) \tag{2.8}$$

This can be rewritten.

$$\frac{l_1 - l_3}{n(\cos(\phi_3) - \cos(\phi_1)} = 2d \cdot \sin(\theta/2) \tag{2.9}$$

We can do the same for l_2 and l_4.

$$\frac{l_2 - l_4}{n(\cos(\phi_4) - \cos(\phi_2))} = 2d \cdot \sin(\theta/2) \tag{2.10}$$

If we use the ϕ_i's as defined in (2.1) and equate 2.9 and (2.10).

$$\frac{l_2 - l_4}{l_1 - l_3} = \frac{\cos(\phi_4) - \cos(\phi_2)}{\cos(\phi_3) - \cos(\phi_1)} = \frac{\sin(\phi)}{\cos(\phi)} = \tan(\phi) \tag{2.11}$$

This can be used to find an expression for ϕ, the angle of curvature.

$$\phi = \arctan(\frac{l_2 - l_4}{l_1 - l_3}) \tag{2.12}$$

The tendon lengths and the radii of curvature are related.

$$\frac{R}{R_i} = \frac{l_c}{l_i/n} \tag{2.13}$$

This yields an expression for R.

$$R = \frac{R_i \cdot l_c}{l_i/n} \tag{2.14}$$

Substituting (2.4) in (2.14), gives us R the radius of curvature.

$$R = \frac{l_c \cdot n \cdot (R - d \cdot \cos(\phi_i))}{l_i} = \frac{R \cdot l_c \cdot n}{l_i} - \frac{l_c \cdot n \cdot d \cdot \cos(\phi_i)}{l_i} \tag{2.15}$$

$$R - \frac{R \cdot l_c \cdot n}{l_i} = R(1 - \frac{l_n \cdot n}{l_i}) = -\frac{l_c \cdot d \cdot \cos(\phi_i) \cdot n}{l_i} \tag{2.16}$$

$$R = \frac{l_c \cdot d \cdot \cos(\phi_i)}{l_c - l_i/n} \tag{2.17}$$

Equations (2.2), (2.12) and (2.17) describe the pose of the continuum robot for arbitrary tendon lengths. In figure 2.5 the continuum robot is shown in the plane of curvature. This plane is the (X, Z) plane that is rotated by ϕ over the Z-axis.

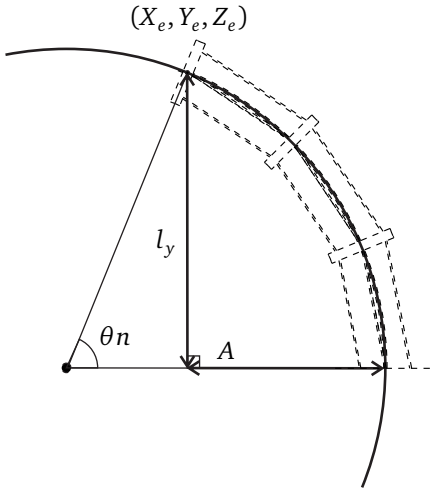


Figure 2.5: A continuum robot shown in its bending plane. This is the (X, Z) plane rotated by ϕ over the Z-axis

We can calculate θn from the radius of the circle and the length of the device.

$$l_b = 2\pi \cdot R\frac{\theta n}{2\pi} => \theta n = l_b/R \tag{2.18}$$

Then we can use trigonometry to calculate A. This is the distance between the current end effector position in the (X, Y) plane in relation to its neutral position.

$$A = R - R \cdot \cos(\theta n) \tag{2.19}$$

Similar for l_y, which is the length of the device in the Z-direction. The robot gets shorter in this direction when it bends.

$$l_y = R \cdot \sin(\theta n) \tag{2.20}$$

Then we can derive the relations for (X_e, Y_e, Z_e). But we need to add the baseplate (described in section 3.2.3) thickness d_{base} and the distance of the base from the origin in the Z direction, l_{lin} to get Z_e.

$$X_e = A \cdot cos(\phi), Y_e = A \cdot sin(\phi), Z_e = l_{lin} + l_y + d_{base} \tag{2.21}$$

Equation 2.21 are the relations we need to calculate the end effector position from the tendon lengths and the linear stage position.

2.3 Inverse Kinematics

In order to steer the position of the end-effector, we need to derive the inverse kinematics. These show the relation between the end effector position to the tendon lengths and linear position, $(X_e, Y_e, Z_e) \rightarrow (l_1, l_2, l_3, l_4, l_{lin})$. As discussed in section 2.2 the continuum robot can be described by radius R and angle ϕ. First, ϕ is determined.

$$\phi = \arctan\frac{Y_e}{X_e} \tag{2.22}$$

In order to find R we first need to calculate A.

$$A = \sqrt{X^2 + Y^2} \tag{2.23}$$

If we combine (2.19) and (2.18) we get the expression to solve needed to get R. It is impossible to get a closed form expression of (2.24). Therefore, this equation has to be solved numerically.

$$A = R - R \cdot \cos(l_b/R) \tag{2.24}$$

The angle θn can be calculated from R and l_b

$$l_b = 2 \cdot \pi \cdot R \cdot \frac{\theta \cdot n}{2 \cdot \pi} \Rightarrow \theta = \frac{l_b}{n \cdot R} \tag{2.25}$$

We can now rewrite (2.3).

$$l_c = 2 \cdot R \cdot \sin(\frac{l_b}{2 \cdot n \cdot R}) \tag{2.26}$$

With this equation and by rewriting (2.7) the individual tendon lengths can be calculated.

$$l_i = n(l_c - 2 \cdot d \cdot \cos(\phi_i) \cdot \sin(\frac{l_b}{2 \cdot n \cdot R})) \tag{2.27}$$

The linear stage must be set to a position based on the Z_e position. But the devices gets shorter as its bends so we need to take l_y and d_{base} in account. So we rewrite (2.21 using (2.20) and (2.25).

$$l_{lin} = Z - R \cdot \sin(n \cdot R \cdot l_b) - d_{base} \tag{2.28}$$

This means we have a mapping from (X_e, Y_e, Z_e) to tendons lengths and linear stage position l_{lin}.

In the next chapter the design process of the system will be discussed.

Chapter 3

Design

In this chapter the design of the continuum robot and its actuator will be discussed. The robot and its actuator are shown in figure 3.1. The decision to make a continuum robot was made for several reasons. The first of which was that because of the continuously bending nature of the continuum robot it is ideally suited for Natural Orifice Surgery (NOS). The absence of any discrete joints will help the robot navigate through the body with little resistance. This means that it is less likely that damage will be done to the body. For navigation *in vivo* the robot needs to be able to change shape precisely. It has been shown that continuum robots are capable of this [12].

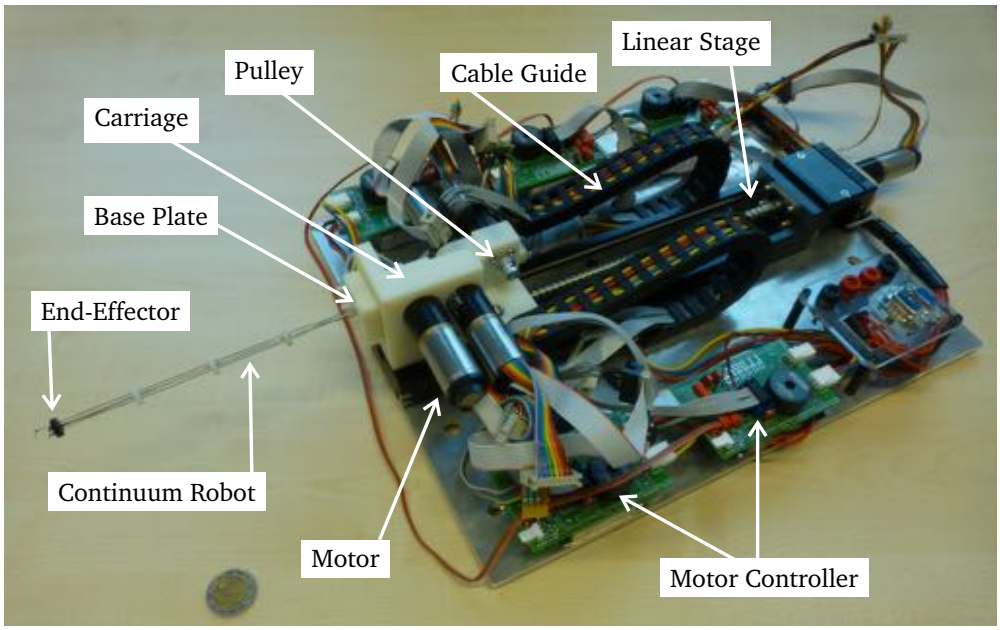


Figure 3.1: The complete system with parts labelled

The complete system had to meet certain requirements. These requirements will be discussed in section 3.1. The design of the continuum robot is presented in section 3.2. The actuator that drives the continuum robot is discussed in section 3.3.

3.1 Requirements

This study was a proof of concept. But since the design needed to have medical applicability, some requirements came from this background. There were also requirements that were dictated by availability of resources at the Medical Robotics Laboratory. The point of this study was to show that a simple kinematic model is sufficient to drive continuum robot when combined with 3D position feedback control. Therefore the design of the continuum robot did not have to be mechanically complicated. These requirements lead to the requirements of the continuum robot, these are presented in section 3.1.1 and the requirements of the actuator are presented in section 3.1.2.

Requirements of the complete system

- Continuum robot
- Simple construction.
- Simple method of actuation

The choice was made to use a continuum robot. A simple construction was preferable. A very complicated design that compensated for the innate shortcomings of these kind of continuum robot would have defeated the purpose of this study. The focus was on performance improvements through the feedback system.

3.1.1 Continuum Robot

Requirements

- Tendon driven
- Fit inside a natural orifice of the body
- Cover a workspace of at least 100 mm^3
- Be able to bend at at least a 45° angle

The design that was chosen was a tendon actuated continuum robot. This decision was made because these are simple to fabricate and pulling tendons is an easily implemented actuation method. The tendon guides could be 3D printed on the available Objet Eden 250. This meant that designs could be fabricated at relatively low cost and with high accuracy, allowing for rapid prototyping. The robot was required to bend at a minimum of 45°. This requirement was made to simulate navigation requirements for the human body. To be able to do this, the backbone needed to be flexible. It was also deemed useful if it had some elasticity to return to a neutral position on its own.

3.1.2 Actuator

Requirements

- Be able to do an insertion motion
- Be able to move the end-effector at $\geq$10 mm/s
- Needs to be flexible enough to drive different models of continuum robot
- Needs to be able to pull with a force of $\geq$20 N

- Fit on a optical breadboard in a space of 300x370 mm

To allow for an insertion motion a linear movement is required. A speed of 10 mm/s was chosen as the minimum speed mimicking the slow methodical movement of a surgeon during an operation. A certain amount of flexibility was required from the actuator, the radius of the continuum robot varied and the actuator should have been able to drive all designs with as little modification as possible. Also the motors had to be pull at least 20 N to drive the 2 planned designs. The physical footprint requirement was due to the limited lab space available for the device.

3.2 Continuum Robot

The type of continuum robot that was chosen is also called a snake-arm robot. It consisted of a flexible backbone with tendon guides fixed to it at constant distances from each other. This design could be miniaturized by decreasing the tendon guide size. Except for the backbone, all the parts were 3D printed.

3.2.1 Backbone

The backbone needed to be flexible. It also should not plastically deform when bended. To meet this requirement a Nitinol (A nickel titanium alloy) rod was chosen as the backbone of the device. Nitinol has the property of superelasticity. This means that it is less likely to plastically deform under bending (within limits, i.e. under 10% deformation). Also, it is bio-compatible, so it is a viable material for a clinical environment. Tests were performed and the 1 mm diameter wires had the right flexibility, where the wire bended easily enough but did not buckle under heavy loads.

The 45° bend and the length of the device resulted into a certain workspace. We can use the equations from section 2.2 to calculate this workspace. The device needed to cover an area of at least 100 mm^3 at a 45° bend. The workspace is determined by A (equation 2.19) and l_y (equation 2.20). These variables define a cylindrically shaped workspace. A length of 160 mm was chosen because of the field of view of the cameras. If we combine equation 2.19 and equation 2.25 we can calculate what the size was of the workspace with this device length.

$$A = \frac{l_b}{\theta_n}(1 - \cos\theta n) \Rightarrow \frac{160}{1/4\pi}(1 - \cos 1/4\pi) \approx 60\text{mm} \tag{3.1}$$

Furthermore l_y could be calculated by combining (2.20) and (2.25).

$$l_y = \frac{l_b}{\theta_n} \cdot \sin\theta n \Rightarrow \frac{160}{1/4pi} \cdot \sin(\theta n) \approx 140\text{mm} \tag{3.2}$$

This defines a workspace of $60^2 \cdot \pi \cdot 140 \approx 150\text{mm}^3$. Which met the requirement.

3.2.2 Tendon Guides

The choice was made to use four tendon guides. The tendon guides keep the tendons close to the backbone of the device. This enabled the device to potentially move through passages. The outer diameter of the guides had to be of a size such that the robot would potentially fit inside a natural orifice of the body. The diameter of the rectum is on average 21 mm [11]. Therefore a design with a tendon guide radius of 10 mm was chosen. To the show possibility of miniaturization a design of a smaller radius (5 mm) was also created.

The tendons should be as far away from the backbone as possible. This reduces the forces needed to manipulate the robot. However, if the holes are too close to the outer rim of the tendon guides the tendon will protrude from the robot when it bends. This is the case because, unlike the backbone, the tendons will run from guide to guide in a straight line. This can be seen in figure 3.2. The distance of protrusion is also dependent on the number of tendon guides on the backbone. If the robot bends at radius R and we take a segment of a continuum robot which has n segments, length l_b/n, outer radius d_o, protrusion radius d_{mid} and tendon radius d. This situation is shown in figure 3.2.

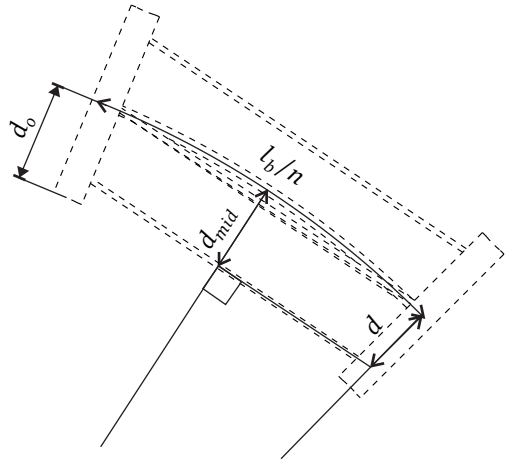


Figure 3.2: A segment of the continuum robot

The relation between the outer guide radius and maximum distance the tendon holes can be placed from the backbone while the tendons do not protrude too far when the device bends can be calculated. As specified in section 3.1.1, the maximum bend the device was going to make is 45°. The tendon protrudes the furthest halfway between the two tendon guides. The tendon may not be further away from the backbone than the outer diameter of the tendon guides, so $d_{mid} \leq d_o$. At this point the tendon guide and the line d_{mid} are at an $\frac{1/4\pi}{2n}$ angle. This gives us a trigonometric relation. With $R-d$ forming the hypotenuse and $R-d_o$ forming the adjacent.

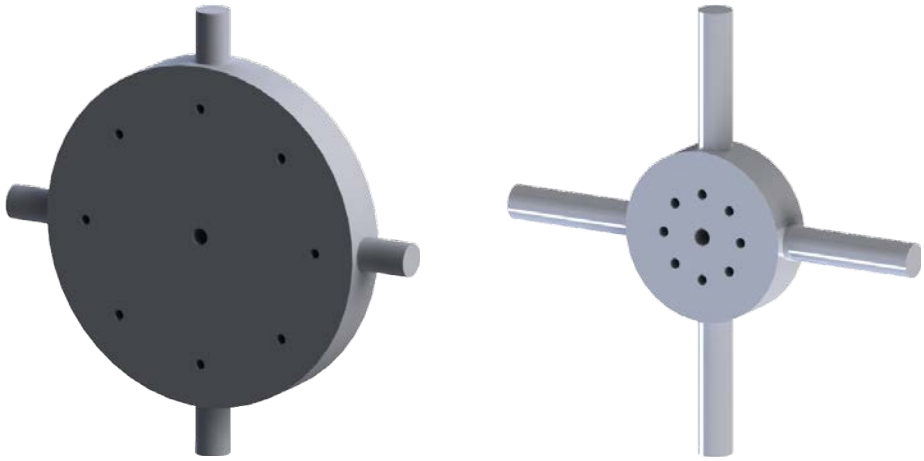
$$\cos\left(\frac{1/4\pi}{2n}\right) = \frac{R-d_o}{R-d} \tag{3.3}$$

We can rewrite this function to get d with $R = \frac{l_b}{\theta n}$ and $\theta n = 1/8\pi$.

$$\frac{\frac{l_b}{\theta n} - d_o}{\cos\left(\frac{1/4\pi}{2n}\right)} = \frac{l_b}{\theta n} - d \Rightarrow d = \frac{l_b}{\theta n} - \frac{\frac{l_b}{\theta n} - d_o}{\cos\left(\frac{1/4\pi}{2n}\right)} \tag{3.4}$$

If we set $l_y = 160$ mm and $n = 4$. This equation yields $d = 8.0$ mm for $d_o = 10.0$ mm and $d = 3.1$ mm for $d_o = 5.0$ mm. However, because of the uncertainty of the strength of the 3D printed material the decision was made to decrease d to 7.5 mm and 2.5 mm respectively.

The guides needed to be fixed to the backbone at constant distances from each other and perpendicular to the backbone. To help achieve this pegs were added to the circumference of the guides. These pegs could be used with the construction aid, which is discussed in 3.2.4. The tendon guides were glued onto the backbone using super glue. The resulting design as shown in figure 3.3.

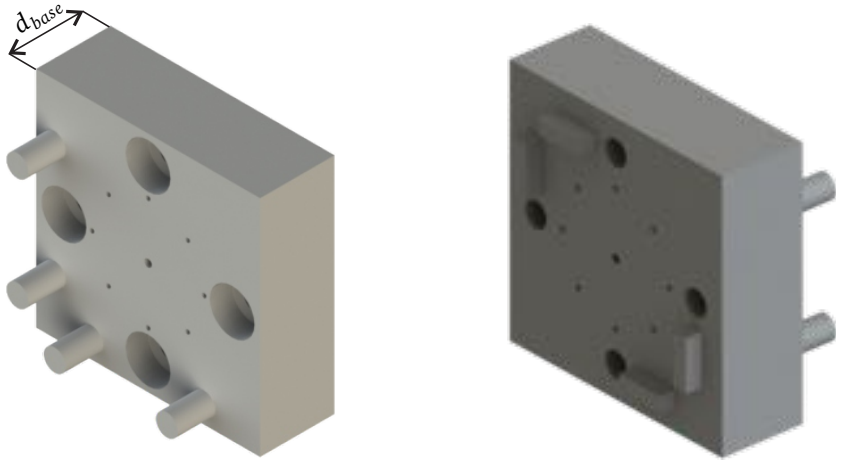


(a) The 10.0mm radius tendon guide (b) The 5.0mm radius tendon guide

Figure 3.3: The different tendon guide designs

3.2.3 Base Plate

The continuum robot needed to be connected to the the actuator. For this a baseplate was designed. The baseplate is shown in figure 3.4. The baseplate thickness d_{base} was chosen at 10 mm to give the robot a solid base where it could be connected to the actuator.



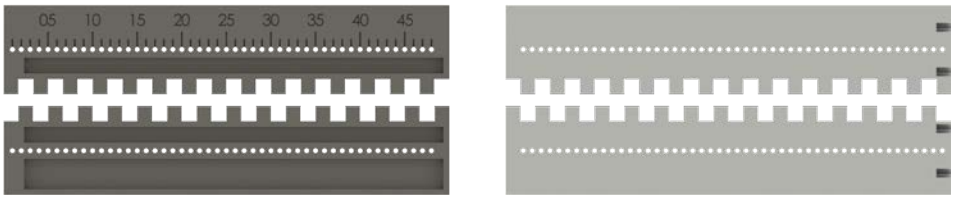
(a) The base plate front side (b) The base plate front (robot) side

Figure 3.4: The different tendon guide designs

The baseplate is a 30x30x10 mm block. It has the same hole pattern as the tendon guides. A number of pegs were added for alignment with the construction aid. There were some holes in the front for M3 bolts so that the base plate could be fixed to the carriage (see section 3.3.1) of the actuator. On the actuator side there were some tapered blocks so that the continuum robot could be attached in alignment with the motors.

3.2.4 Construction Aid

The construction aid was used for making sure the tendon guides and the base plate are attached properly to the backbone, i.e. that they were perpendicular to the backbone, equidistant from each other and in the same orientation. It was built up from two parts as shown in figure 3.5



(a) The outside (b) The Inside

Figure 3.5: Construction aid

The tendon guides and the base plate could be placed inside this construction aid. The two plates together formed the aid as shown in figure 3.6.

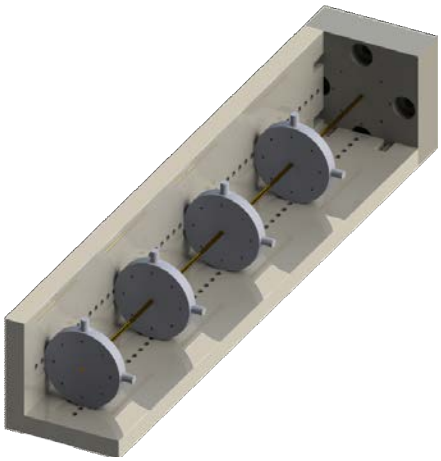


Figure 3.6: The construction aid with the continuum robot

3.3 Actuator

To drive the continuum robot an actuator was constructed. It consisted of several different parts. 3.3.4. The carriage was the part of the actuator to which the continuum robot was attached. Section 3.3.1 discusses the carriage. The motors that pull the tendons were connected to this carriage. The motors are discussed in section 3.3.2. Fixed to the motors were pulleys to which the tendons are connected. The pulley design is discussed in 3.3.3. For an insertion motion the actuator had a linear stage. The linear stage is discussed in section All the parts are listed in Appendix B and all CAD drawings are in Appendix C.

3.3.1 Carriage

The carriage connected the continuum robot to the linear stage and is a 3D printed object. It was designed to be as small as possible. The carriage is shown in figure 3.7

Figure 3.7: The carriage where the continuum robot is connected to the actuator

The carriage had cable guides connected to it for the motor wires and encoder cables. There were two different carriage designs, one for the 5 mm radius tendon guides and one for the 10 mm tendon guides. They were the same, except for the placement of the motors. This was done to make sure that the pulleys line up with the holes in the base plate. This alignment was done to reduce friction. There were grooves in the front of the carriage for alignment of the base plate. There were additional holes in the carriage to make sure all the motor screws could be reached. The carriage complete with motors, continuum robot and cable guides is shown in figure 3.8

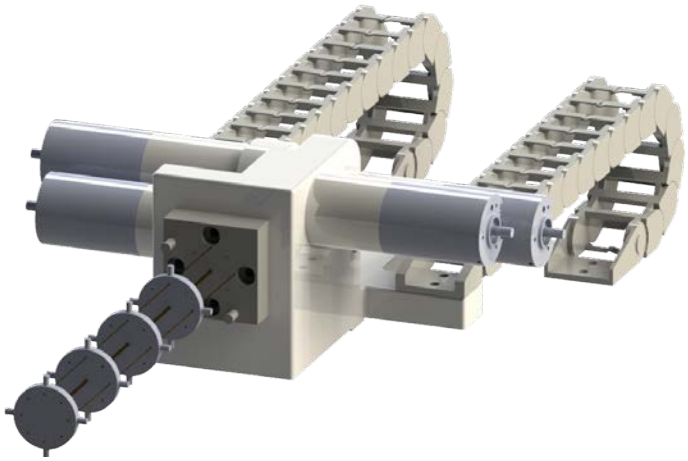


Figure 3.8: The carriage with continuum robot, motors and cable guides

3.3.2 Motors

To determine how strong the motors needed to be a experiments were performed with specials tendon guides that contained multiple holes at certain intervals, d=[2.5 3.75 5.0 6.25 7.5] mm. In these experiments the baseplate was clamped and weights were hung from the tendons. The weight needed to bend the smallest (d=2.5 mm) robot was measured to be $\sim$2 kg or a force of about $2 \cdot 9.81 \approx 20$ N. This is the force required to keep the robot in place, the robot was also required to move at a speed of at least 10 mm/s. If we assume that the robot accelerates linearly the end-effector needed to accelerate at 20

mm/s^2 to cover 10 mm in 1 second. The bigger the deflection of the robot the further the tendons needed to be pulled for the end-effector to move the same distance. It has been calculated in section 3.2.1 that at 45° the deflection is about 60 mm. We need to solve equation (2.23) for $A = 60$ mm and $A = 50$ mm. This gives us $R \approx 202$ mm and $R \approx 247$ mm. We can now solve equation (2.27) to get the difference in tendon length. This is a difference of about 0.44 mm. So for the tendons to be pulled 0.44 mm in 1 second. If we assume linear acceleration the acceleration required is 0.88 mm/s^2. This gives us a total force of $2 \cdot 9.81 + 2 \cdot 0.88 \cdot 10^{-3} \approx 20$ N. The pulleys have a radius of 5 mm (see section 3.3.3) so the motors had to be capable of a torque of $20 \cdot 5 \cdot 10^{-3} \approx 100$ mNm.

As specified in section 3.1.2 the continuum robot needed to move at 10 mm/s. Near 45° bending the tendons needed to be pulled the furthest to move the end effector 10 mm. The d=7.5 mm design was the one that needs to shorten the tendon by the most length to create a certain bend. Therefore, to see at what speed the motors need to be capable we needed to find the difference between l_i's for $A = 60$ mm and $A = 50$ mm for the 10 mm design, this is 1.15 mm. To pull the tendons 1.15 mm the pulley has to turn $\frac{1.15}{5}/2\pi \cdot 360 \approx 120°$ which is 1/3 rotation. To do this in one second the pulley needed to turn at 2/3 rotations per second (Assuming linear acceleration and deceleration). So the motor-gearbox combination needed to be capable of 40 rpm.

The motors are all the same, the Maxon Motor EC-Max 22. This motor was chosen because they fitted the specifications. It can deliver a continuous torque 10.8 mNm and nominal speed of 8260 rpm. It was coupled with a 128 gear reduction maximize torque and reduce speed. This gearbox had an effiency of 59%. Therefore the maximum continuous torque was $10.8 \cdot 128 \cdot 59\% = 815$ mNm. This is enough to drive the continuum robot. The speed goes down to approximately 65 rpm, enough to drive the 10mm design within specifications. This meant that should the design change, the motors could still be used.

The motors were connected to Elmo SimpliQ Whistle 2.5/60 controllers. This controller was chosen because it was capable of controlling the Maxon motors. They can deliver 2.5 A of continuous current, enough to drive the motors which required a peak-current of 1.97 A. The motor controllers were set to position control mode. The pulley motors had a gearbox reduction of 128 and the encoder was a 512 pulses per rotation quadrature encoder. This meant that the pulleys could be set to an arbitrary angle θ_i by setting an encoder position enc_i.

$$\theta_i = enc_i \cdot \frac{1}{2048} \cdot \frac{1}{128} \cdot 2\pi \tag{3.5}$$

3.3.3 Pulley

The pulleys connected the tendons to the motor. The pulleys were designed to be as small as possible. The tendons were pulled as close to the gearbox as possible to not exceed the maximum allowable radial load of the gearboxes. This is specified at 70 N halfway the gearbox shaft (5 mm from the gearbox), i.e. 350 mNm. The pulley design means that the tendon was pulled 7 mm from the gearbox. Resulting in a torque of 140 mNm. So the load was within specifications. The pulleys are shown in figure 3.9.

(a) The outside (b) The motor side

Figure 3.9: The Pulleys

There were two holes in the pulley, one for the set screw to bolt the pulley to the gear axle and one for the tendon inside the groove. The pulleys clamped the tendon between the motor flange and the set screw as shown in figure 3.10

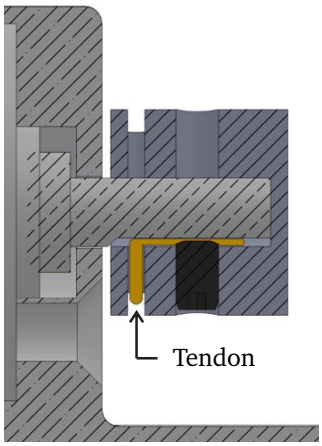


Figure 3.10: The tendon clamped inside the pulley

The pulleys had a radius of 5 mm. It was assumed that the tendon lengths when the robot is straight were equal to l_b, the backbone length. This gives a relation between the pulley angles and the tendon length.

$$l_i = l_b - \theta_i \cdot 5 \tag{3.6}$$

3.3.4 Linear Stage

The linear stage consisted of a linear actuator, a motor and gearbox. The linear stage was a Misumi LX30. This is a single axis linear actuator to which a motor can be connected. It has a small sled on which the carriage was bolted. This linear stage had an effective

stroke of 204 mm. The requirements specified that the end effector needed to move at 10 mm/s. The linear actuator had a spindle with a lead of 5 mm per rotation. This meant that the motor gearbox combination needed to turn at 120 rpm. The motors were the same model as used for pulling the tendons. This motor did not have to deliver any amount of torque. A 19x gearbox reduction was chosen. This reduction gave a maximum speed of approximately 430rpm. Enough to drive the linear stage. The linear stage had a simple relation between motor angle and linear position l_{lin}. The lead of the spindle is 5 and the gearbox had a 19 time reduction so from encoder position enc_{lin} to l_{lin}

$$l_{lin} = \text{enc}_{lin} \cdot \frac{1}{2048} \cdot \frac{1}{19} \cdot \frac{5}{1} \tag{3.7}$$

The LX30 is shown in figure 3.11.

Figure 3.11: The LX30 linear stage

In the next chapter the controller is discussed.

Chapter 4

Control

The focus of this study was on combining a simple kinematic model with a feedback controller. This chapter discusses the controller. The controller was a modified PID-controller. A normal PID-controller was not sufficient because of the excessive delay in the feedback system as described in section 4.1. The PID-controller had a modified I and D action as is discussed in section 4.2. The entire controller ran at 15Hz. The kinematic model was used to drive the system directly in case of measurement failure.

4.1 Feedback System

The feedback system consisted of a dual camera setup. Both cameras are SONY XCD-SX90's. It can be seen in figure 4.1.

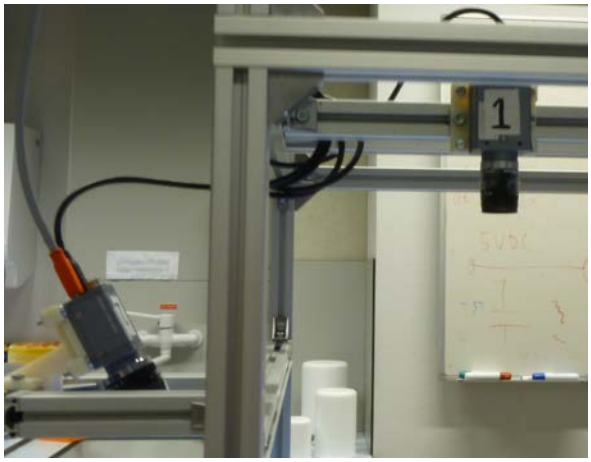


Figure 4.1: The Dual camera setup used for the 3D positioning

These cameras were connected to a Windows computer via firewire. These took 8-bit greyscale images at a resolution of 1280x960 pixels. The cameras were mounted on a frame that was bolted to an optical breadboard. The cameras were mounted using 3D printed mounts to make sure they were securely fastened in the correct position. These mounts consisted of two parts, one mounted to the frame and the other one to the camera to make sure their alignment was consistent. The cameras were at a 45° and 60° angle in relation to each other as can be seen in figure 4.1. A back light was placed underneath

the working area of the robot to make sure the contrast was high enough to discern the tendon guides. The end-effector position was found in the images by the algorithm discussed in section 5.1.2. These two position were then combined to triangulate a 3D position by the triangulation algorithm described in section 5.1.3. This camera setup had a sub millimetre resolution: ~0.1 mm. The cameras were synchronised through the firewire bus. The available bandwidth limited their framerate to 15 fps. This meant the entire controller was limited to 15 fps.

Camera systems are at least one frame behind the input when computing the error. However, experiments have shown that in this case the delay was much higher than one frame. The delay was 5 frames, this was probably due to the firewire implementation on Windows. This delay was significant for tuning the controller.

4.2 Modified PID-feedback Control

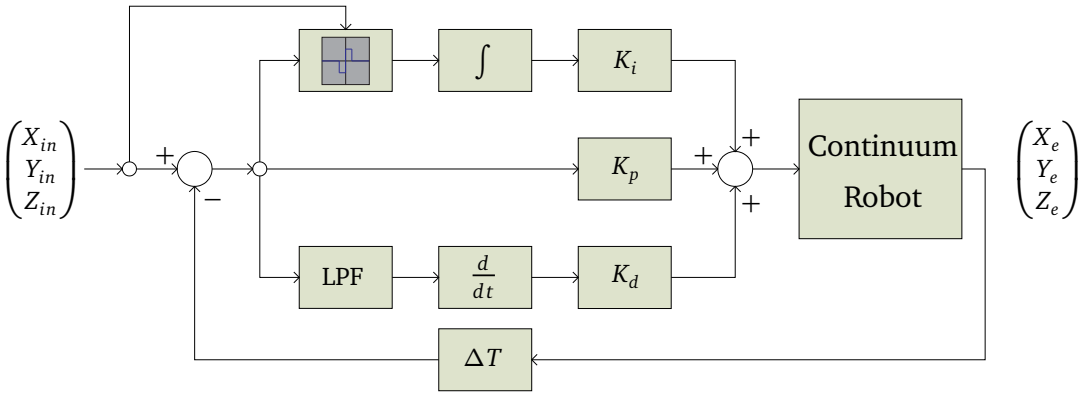


Figure 4.2: Schematic of the Modified PID Controller

The controller as shown in figure 4.2 is a modified PID-controller. This modification was done to compensate for non continuous inputs. The camera setup provided the (X_e, Y_e, Z_e) feedback signal. There is a ~333ms delay in the feedback line. The controller was tuned by experimentation, increasing the gains until the system became unstable and then reducing the gains to get enough margin.

4.2.1 P-Action

The P-action was a normal proportional controller. K_p is a vector of length 3. It had a different gain for each spatial dimension. Through experimental optimization K_p was determined. Controller gains of $K_{px} = K_{py} = K_{pz} = 1.0$ yielded a stable response.

4.2.2 I-Action

To compensate for the delay the I-action does not work until the end effector is within 10% of the input. This was because otherwise a large error would accumulate due to the delay introduced by the delay in the feedback signal. Again, K_i was a vector of length 3. Controller gains of $K_{ix} = K_{iy} = K_{iz} = 1.0$ yielded a stable response.

4.2.3 D-Action

The D-action of the control had a low pass filter in front of it. This filter rejected changes bigger than half the current input signal in the error signal. This was done to increase the stability when a non-continuous input was applied. Gains of $K_{dx} = K_{dy} = K_{dz} = 0.25$ yielded a stable response.

In the next chapter the software is discussed.

Chapter 5

Software

The program that ran the controller and drove the continuum robot was a multi-threaded C++ win32 application. It ran comfortably at 15Hz on a Intel Xeon E5520 with 12GB ram. The program was split up in several threads because of performance and extendibility. All these threads performed one individual task. These tasks are described in section 5.1. They exchanged data through shared resources. The entire system ran at 15Hz even though some threads ran at a higher loop time. The system had four inputs and one output. The input could be switched between the Omni Phantom haptic device for human input and a programmed pattern or step responses as can be seen in figure 5.1. Also, the two cameras were connected to the system. The only output of the system was the CAN bus. The CAN bus connected the system to the five Elmo motor controllers.

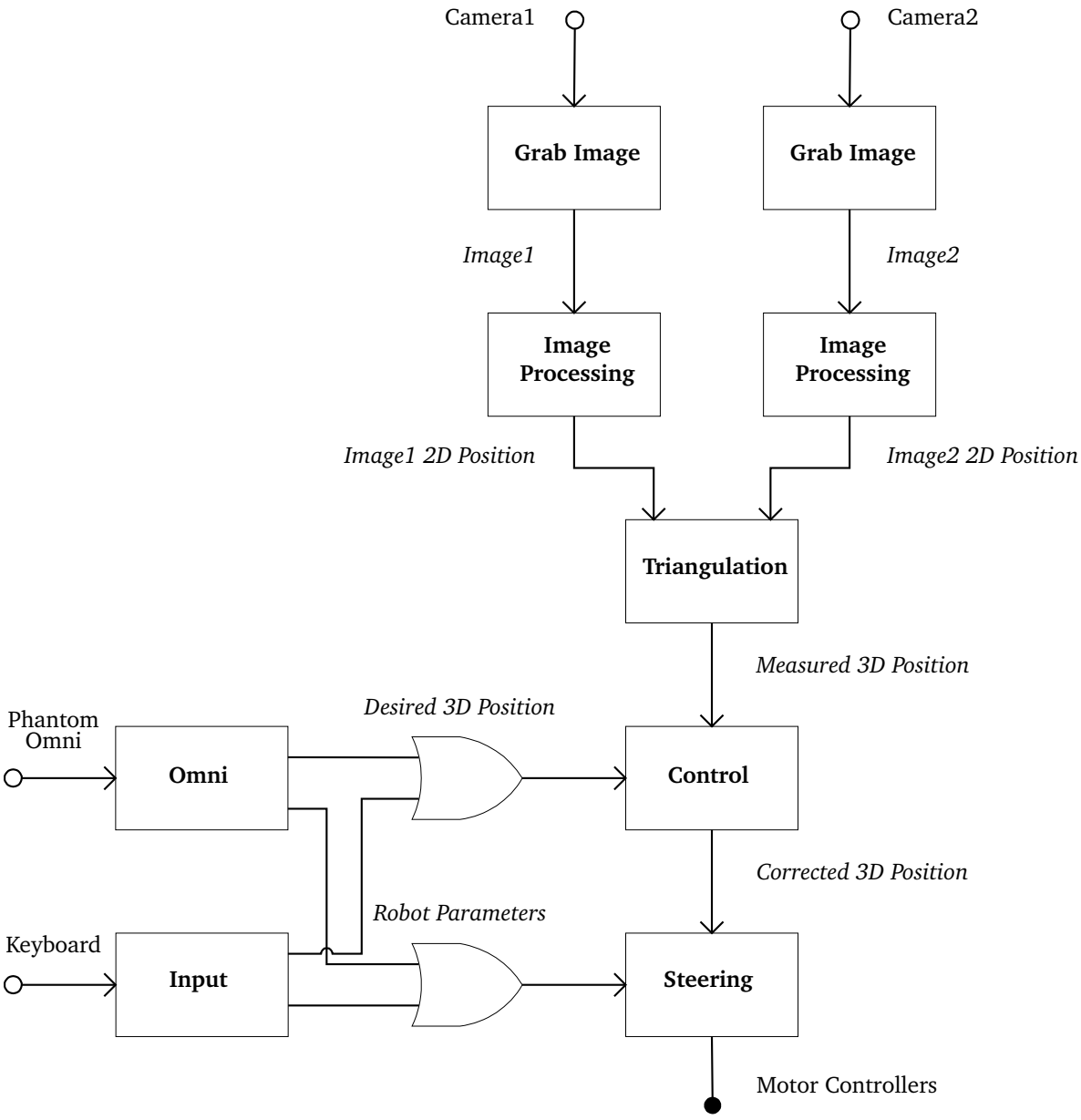


Figure 5.1: Software thread overview with inter-process dataflow

5.1 Threads

As shown in figure 5.1 the software was built up out of different threads. This section describes the function of these threads.

5.1.1 Camera

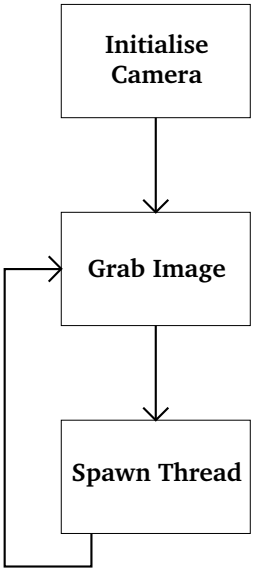


Figure 5.2: Flow of the Camera Thread

The camera thread flow can be seen in figure 5.2. It started the camera with the correct settings and started grabbing images. For communication with the camera the CMU 1394 Digital Camera Driver (`http://www.cs.cmu.edu/~iwan/1394/`) was used. Every time the thread grabbed an image it spawned an image processing thread.

5.1.2 Image Processing

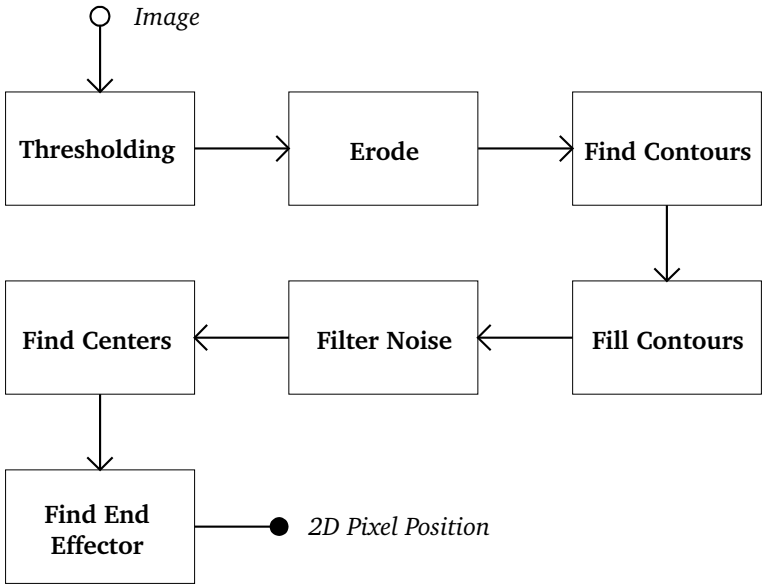


Figure 5.3: Flow of the Image Processing Thread

The image processing threads as can be seen in figure 5.3 find the end-effector in the image. It did this in several steps. The input was a 1280x960 8-bit greyscale image. The image processing was done using functions from the Open Source Computer Vision (OpenCV : `http://opencv.willowgarage.com/`) library. Of the image only the region of interest was processed. This region of interest (ROI) as well as the tuning of the different steps in this process were done in the camera calibration as explained in appendix A. In the first step the ROI was thresholded. Due to a backlight placed beneath the robot the image was bright except where the continuum robot was. Everything that was too bright was rejected. The result was eroded to remove the backbone. The results of these steps are shown in figure 5.4

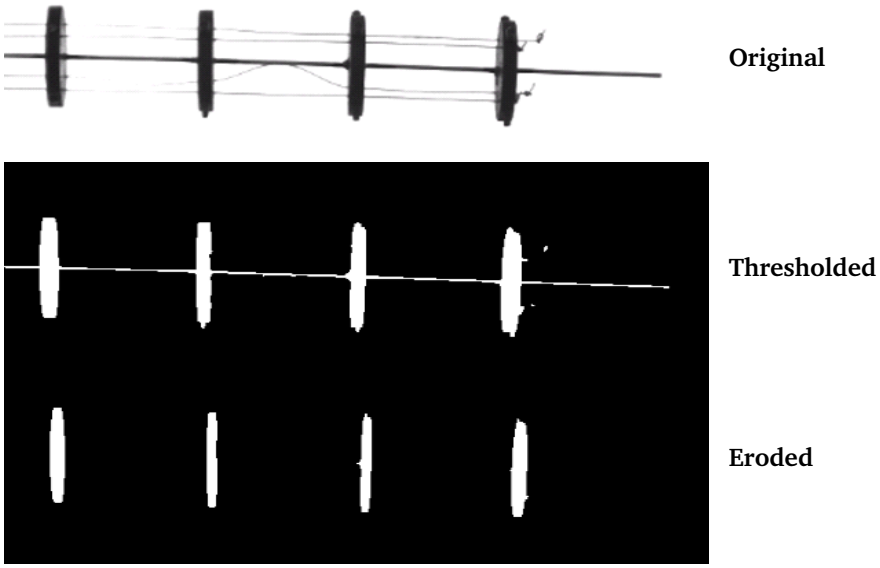


Figure 5.4: Image processing steps

The algorithm then found the different shapes in the image and filled in these shapes. This was done so that the center of masses of the shapes could properly be established. The shapes that were too small to be tendon guides were rejected to reduce noise. After this the center of masses of the blobs were found. The shape the furthest away from the origin of the system is the end-effector as the device did not bend more than 90°. This yielded a result in pixel coordinates for the center of the end effector, this information was set for the triangulation thread to process.

5.1.3 Triangulation

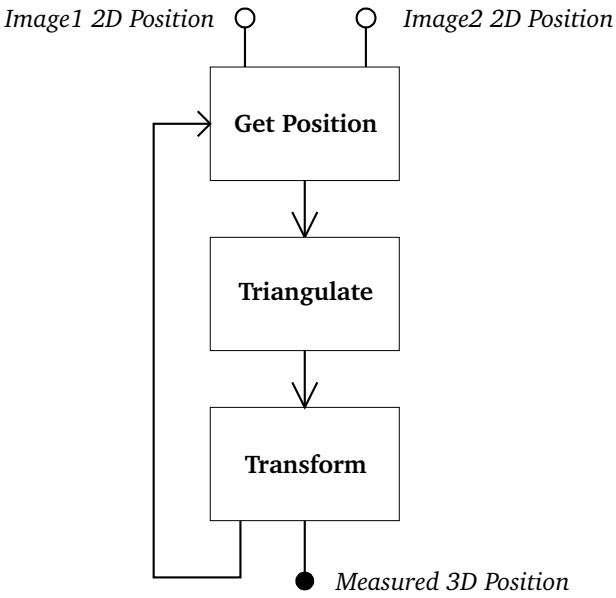


Figure 5.5: Flow of the Triangulation Thread

The triangulation thread, as can be seen in 5.5 took the position of the end effector in both the images in 2D pixel position and triangulated the 3D position in real world coordinates. The triangulation was done by a method from Hartley et al. [8]. This algorithm uses the camera matrices and the rotation and translation between the cameras as calculated by the camera calibration as described in appendix A. These camera calibration matrix describe how a point in real world parameters is projected unto the image. This relation ($\boldsymbol{C \cdot X_c = x_i}$) is shown in (5.1).

$$\begin{pmatrix} f_x & 0 & pp_x \\ 0 & f_y & pp_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \begin{pmatrix} X_c \cdot f_x + Z_c \cdot pp_y \\ Y_c \cdot f_y + Z_c \cdot pp_x \\ Z_c \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ Z_c \end{pmatrix} \tag{5.1}$$

In this relation f_x and f_y are the focal lengths in pixels in the relevant directions. Likewise, pp_x and pp_y is the principal (center) point of the camera image in pixels. The position (X_c, Y_c, Z_c) here are expressed in the camera frame. On the right hand side x_i and y_i are the pixel positions. We can define projection matrices $\boldsymbol{P}$ for both cameras. Using these projection matrices the position can be expressed in the same frame. In this case the frame of camera 1 was chosen as a common frame. The stereo calibration gives us the rotation matrix R and the translation vector T between these two cameras. This means that $\boldsymbol{P_1 = C_1}$ and $\boldsymbol{P_2 = C_2 \cdot (R|T)}$.

The DLT algorithm from Hartley and Zisserman [7] was used. This algorithm yields a set of 4 equations that needed to be solved to triangulate the end-effector. These equations

are shown in (5.2) in the form $\boldsymbol{A} \cdot \boldsymbol{X_C} = \boldsymbol{B}$.

$$\begin{pmatrix} x1_i \cdot P1_{3,1} - P1_{1,1} & x1_i \cdot P1_{3,2} - P1_{1,2} & x1_i \cdot P1_{3,3} - P1_{1,3} \\ y1_i \cdot P1_{3,1} - P1_{2,1} & y1_i \cdot P1_{3,2} - P1_{2,2} & y1_i \cdot P1_{3,3} - P1_{2,3} \\ x2_i \cdot P2_{3,1} - P2_{1,1} & x2_i \cdot P2_{3,2} - P1_{1,2} & x2_i \cdot P2_{3,3} - P2_{1,3} \\ y2_i \cdot P2_{3,1} - P2_{2,1} & y2_i \cdot P2_{3,2} - P2_{2,2} & y2_i \cdot P2_{3,3} - P2_{2,3} \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} -x1_i \cdot P1_{3,4} - P1_{1,4} \\ -y1_i \cdot P1_{3,4} - P1_{2,4} \\ -x2_i \cdot P2_{3,4} - P2_{1,4} \\ -y2_i \cdot P2_{3,4} - P2_{2,4} \end{pmatrix} \tag{5.2}$$

This equation could not be solved exactly because of the difference between the perceived points in the cameras and the inaccuracies in $\boldsymbol{C}$, $\boldsymbol{R}$, and $\boldsymbol{T}$. OpenCV includes a matrix solver, this was used with a singular value decomposition (SVD) method.

Because the location of the origin and the rotation to the origin was established in the calibration this knowledge is used to transform (X_c, Y_c, Z_c) to the robot frame (X_e, Y_e, Z_e).

5.1.4 Control

The control thread implements the control scheme as described in section 4.2.

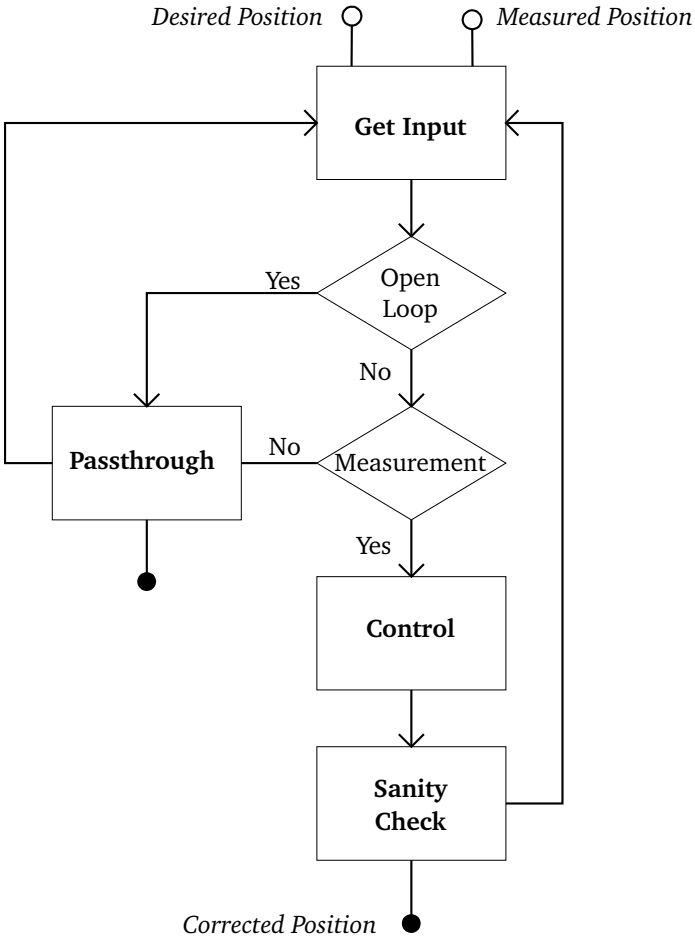


Figure 5.6: Flow of the Control Thread

The controller as shown in 5.6 worked at 15Hz. As input it has the current desired position and the measured position. If the controller was set to open loop the desired position is set as the corrected position. If it was set to close-loop a check was made whether a measurement has actually been made, if not the desired position was fed unchanged to the steering thread. If a correct measurement had been made a control step was performed. The output was checked whether was not beyond specifications and might break the robot in case of errors.

5.1.5 Steering

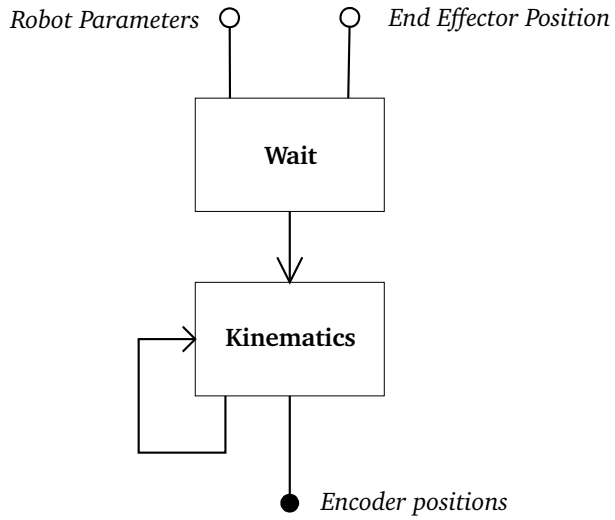


Figure 5.7: Flow of the Steering Thread

The steering thread as shown in 5.7 used the model described in section 2.3 to set the encoder positions. It waited until the current robot parameters such as number of tendon guides, tendon hole radius and robot length were set before starting.

5.1.6 Input & Omni

The input and omni threads both allowed the user to set the robot paramaters: length, tendon guide radius and the number of tendon guides. After this the input tread either asked the user for a position or outputs a programmed sequence of positions at 15 Hz. The Phantom Omni is a haptic device made by Sensable (`http://www.sensable.com/haptic-phantom-omni.htm`), it is shown in figure 5.8.

Figure 5.8: The Sensable Phantom Omni

It allowed the user to point at something in 3D. The Omni could also deliver force feedback. When the haptic device was chosen as the input of the system. The user could point the stylus at a point in space and the actuator positioned the robot in the same place (in its own coordinate system obviously). If the user moved the omni beyond the workspace of the robot a force was generated to alert the user of this.

The next chapter discusses the experiments that were performed to establish the performance of the device.

Chapter 6

Experiments

To compare the performance of the system between open and closed loop operated several experiments were performed. The experiments consisted of providing several different input function to both the open-loop and closed-loop systems.

The experiments were done by programming different functions into the the input thread (see section 5.1.6). These experiments were done using the 10 mm radius tendon guide design. The camera setup was used as a measurement tool. The time delay in the camera setup was corrected when evaluating the performance. Graphs of the open-loop results are shown in section 6.1 and of the closed-loop results in section 6.2. Open-loop and closed-loop results are quantified and compared in section 6.3. The human input with the Sensable Phantom Omni was also evaluated. The results from this are discussed in 6.4. The robot with the smaller tendon guide size (5 mm radius) was also tried, these results are in section 6.5. In all these experiments the robot was pre-tensioned so that a 1° angle change of the pulleys resulted in a measurable change in position of the end effector.

6.1 Open Loop

In the open loop case the tendon lengths were determined by the kinematic model from chapter 2. These experiments were done to determine the quality of the kinematic model. So we could see whether the closed-loop setup actually improved performance and by how much. First some step responses were applied and then some general geometrical shapes.

6.1.1 Step Responses

The step responses were done by requesting a 60 mm step in end position in one or two directions while the other direction(s) had to remain at zero. The robot moved to this position as quickly as possible. The robot should be capable of at least 10 mm/s so it should be able to complete this movement in 6 seconds. Therefore, a fixed measurement time of 6 seconds was chosen. The Z direction was not tested because the LX-30 linear stage has a precision exceeding the measurement system. Because movement in the X and Y- direction influenced the Z direction, it is still plotted. All experiments were repeated 5 times.

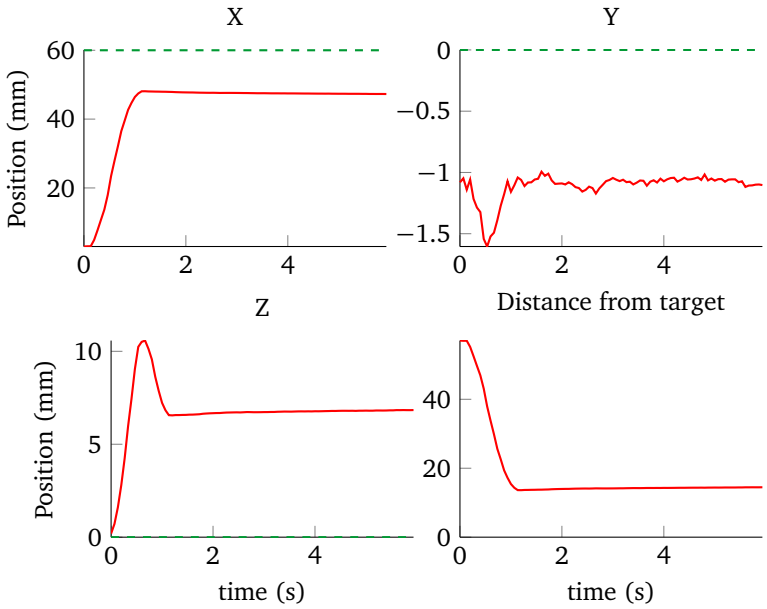


Figure 6.1: The open-loop step response over the X-axis (input in dashed green, output in solid red)

As can be seen in figure 6.1 the kinematic model was capable of driving the robot. However, the end-effector was unable to reach the target. This was probably due to the limited stiffness of the tendons, they stretched when they were pulled. The end-effector missed its target by on average 14.5 mm, the deviation in this result was 0.1 mm. There was also some movement in the Y-direction, this is probably due to some friction. Also, the robot hanged in the Y-direction due to the limited stiffness of the backbone and gravity acting on the device. The model overcompensated in the Z-direction because of the reduced bending the robot did not shorten as much as expected. The tendon also seemed to settle over time, the end-effector moved away from its initial position even though the pulleys had stopped turning.

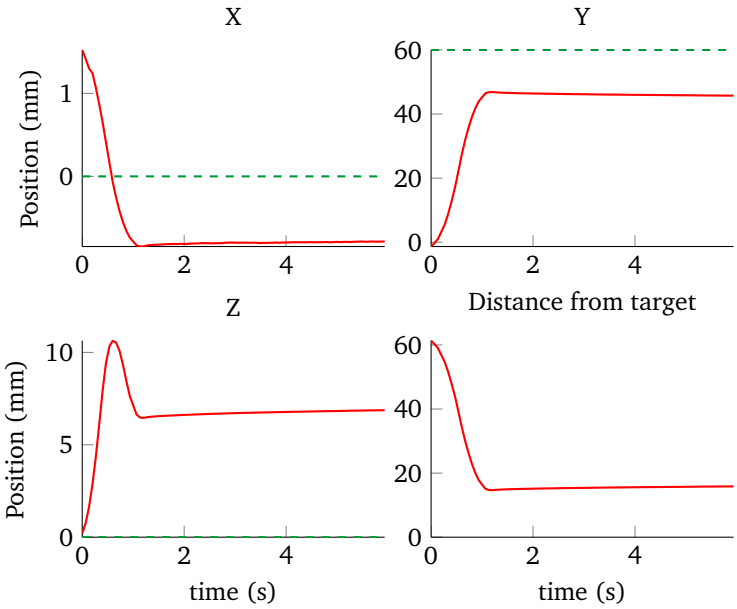


Figure 6.2: The open-loop step response over the Y-axis (input in dashed green, output in solid red)

An input of a step in the Y direction resulted in a response as shown in figure 6.2. The result was similar to the X step result, the end-effector position was off on average by 15.8 mm, with a deviation of 0.1 mm. The device also moved in the X direction probably due to a fabrication imperfection, misalignment of the tendon guides. The Z direction was still overcompensated.

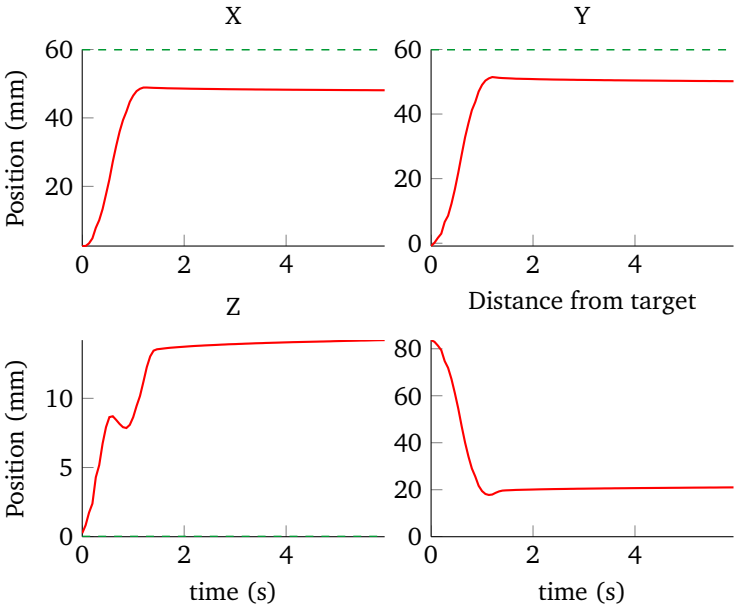
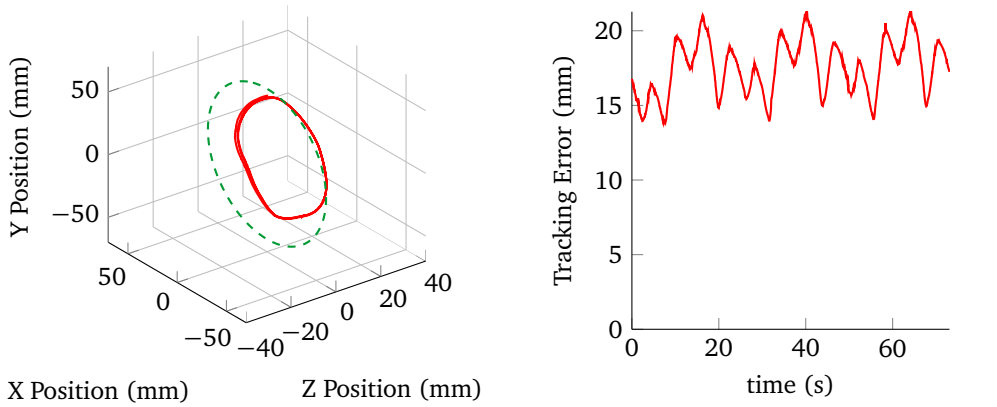


Figure 6.3: The open-loop step response over the X and Y-axis (input in dashed green, output in solid red)

Figure 6.3 shows the result of a combined X and Y step input. This resulted in a response that amplified the previous shortcomings. Because the device had to bend further than in the previous cases the overcompensation over the Z is even high. The end effector misses its target by a higher margin, on average 21.0 mm with a deviation of 0.3 mm.

6.1.2 Shapes

The kinematic model was also tested with some simple geometric shapes. The first of which was a 15 degree/s circle with radius of 60 mm. This means that the end-effector moved at ~15mm/s, about 50% beyond the minimum speed requirement set out in section 3.1. The response in 3D can be seen in figure 6.4a. The result was a slightly deformed circle with a radius smaller than 60 mm. Because of the overcompensation the circles were made in the wrong place Z wise. The error can be seen in figure 6.4b. The error was the largest when both the deflection in X and Y direction were equal, i.e. when the most tension was put on the tendons. The tracking error was on average 17.6 mm with a deviation of 1.8 mm and a maximum of 21.3 mm.

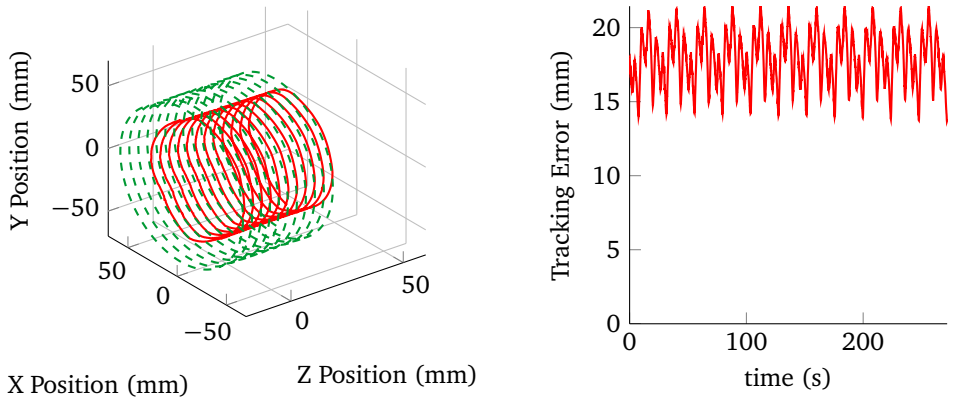


(a) 3D plot (input in dashed green, output in solid red)

(b) The distance between the end-effector and target

Figure 6.4: The open-loop response to a circle shaped input

This experiment was followed by a cylinder shape. This result can be seen in figure 6.5. This shows that the result was very repeatable and similar to the circle shape. The average tracking error was 17.9 mm with a deviation of 1.7 mm and a maximum of 21.3 mm.

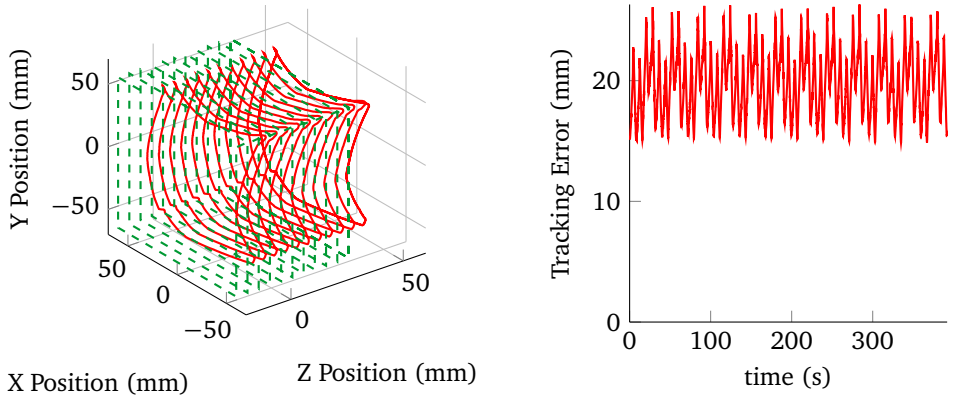


(a) 3D plot (input in dashed green, output in solid red)

(b) The distance between the end-effector and target

Figure 6.5: The open-loop response to a cylinder shaped input

A repeated square (beam) shaped input (multiple 60x60 mm squares) further showed the shortcomings of the kinematics model. This response is shown in figure 6.6. In this case the end effector moved at 15 mm/s. The square was pillow shaped. In this case the curvature of the robot was not constant and therefore the overcompensation in Z direction was not constant. The average tracking error was 17.9 mm with a deviation of 1.7 mm and a maximum of 21.3 mm.



(a) 3D plot(input in dashed green, output in solid red)

(b) The distance between the end-effector and target

Figure 6.6: The open-loop response to a beam shaped input

6.2 Closed Loop

The same experiments were repeated with the closed loop controller from chapter 4. The gains were $K_p = 1, K_i = 1, K_d = 0.25$.

6.2.1 Step Responses

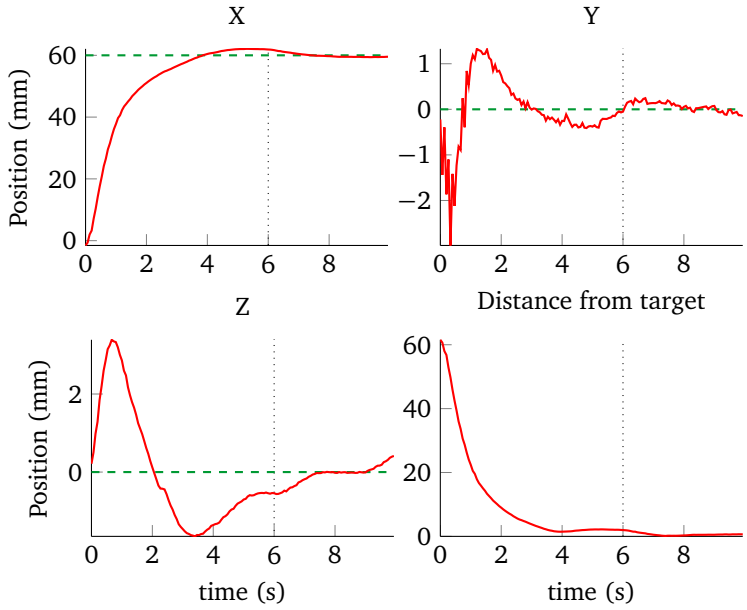


Figure 6.7: The closed-loop step response over the X-axis (input in green, output in red)

The X step results are shown in figure 6.7. The response was slower than the open-loop response. However, the end-effector was closer to the desired position at 6 seconds. The average tracking error was 2.1 mm with a deviation of 0.2 mm and an overshoot of 2.2 mm. The sagging of the device due to gravity was also compensated. The movement in the Y-direction due to friction was compensated for by the feedback controller. The overcompensation in the Z direction was eliminated. The end result oscillated around the desired position but it settled over time. This effect was exaggerated due to the time delay in the feedback signal. Reducing this time delay would result in better performance.

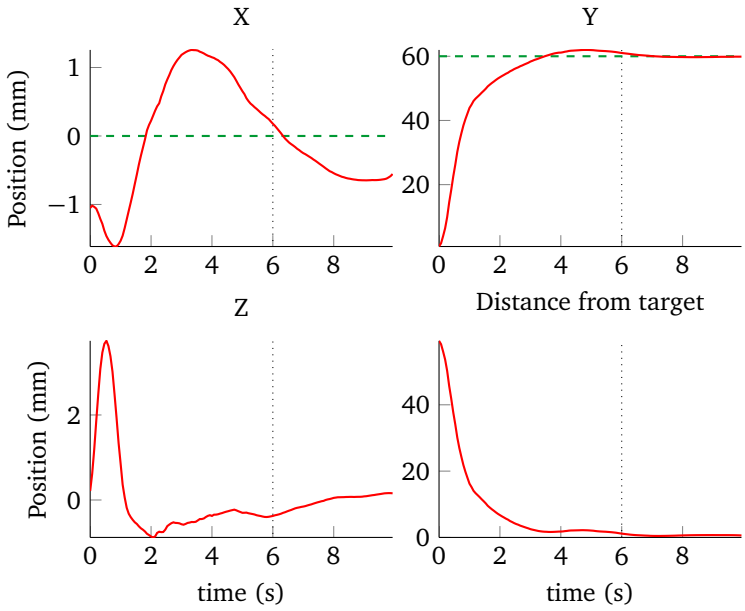


Figure 6.8: The Closed-loop Step response over the Y-axis (input in dashed green, output in solid red)

The Y-step result was similar as can been seen in figure 6.8. Again, the end-effector was closer to the desired position at the 6 second mark. The movement in the X-direction due to the fabrication imperfection is now compensated. The average tracking error was 1.4 mm with a deviation of 0.2 mm and an overshoot of 2.2 mm.

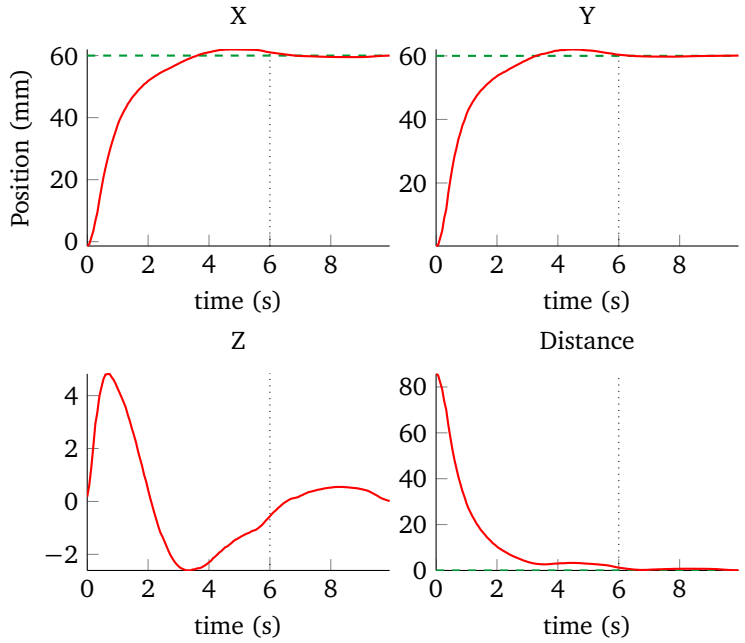
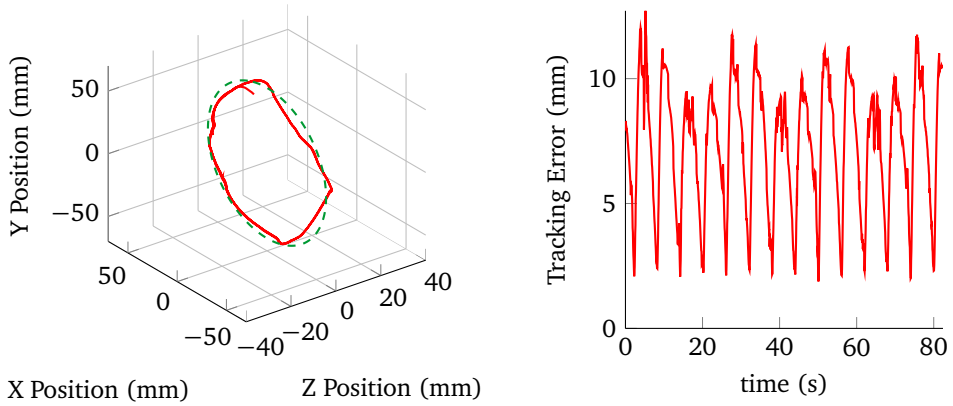


Figure 6.9: The Closed-loop Step response over the X and Y-axis (input in dashed green, output in solid red)

In the closed-loop case when the X and Y step were combined the performance did not deteriorate as in the open loop case. This can be seen in 6.9. The robot got close to the desired position at 6 seconds. The average error was 2.0 mm with a deviation of 0.5 mm and an overshoot of 2.2 mm. There was no overcompensation in the Z-direction.

6.2.2 Shapes

The geometric shapes were also used as input with the closed-loop controller. Here the time delay in the feedback signal became apparent and the gain in performance was not as pronounced as with the step result where the input was constant. The results were better than the open loop case as can be seen in figure 6.10. In this figure the circle shape response is shown. The end effector position was closer to the desired position than with the open-loop system. The circle is still malformed, but less so than with the open-loop system. The end-effector had an average tracking error of 7.4 mm with a deviation of 2.6 mm and a maximum of 12.8 mm.

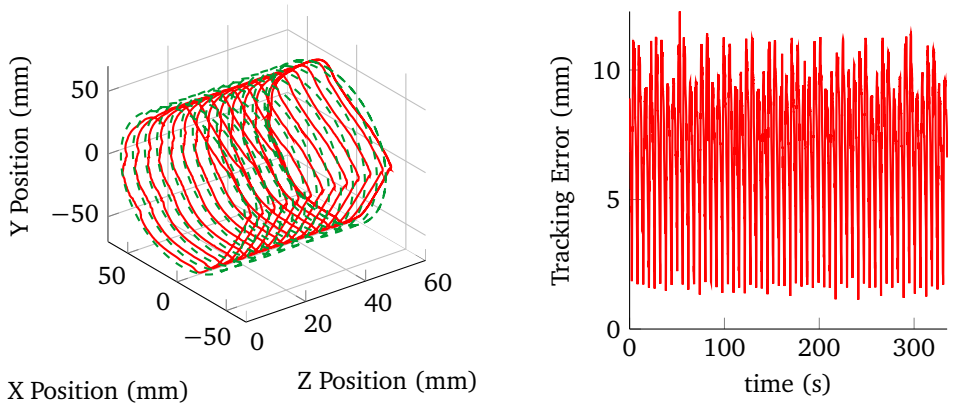


(a) 3D plot(input in dashed green, output in solid red)

(b) The distance between the end-effector and target

Figure 6.10: The closed-loop response to a circle shaped input

Figure 6.11 shows that the results were repeatable. The circles were the same shape.

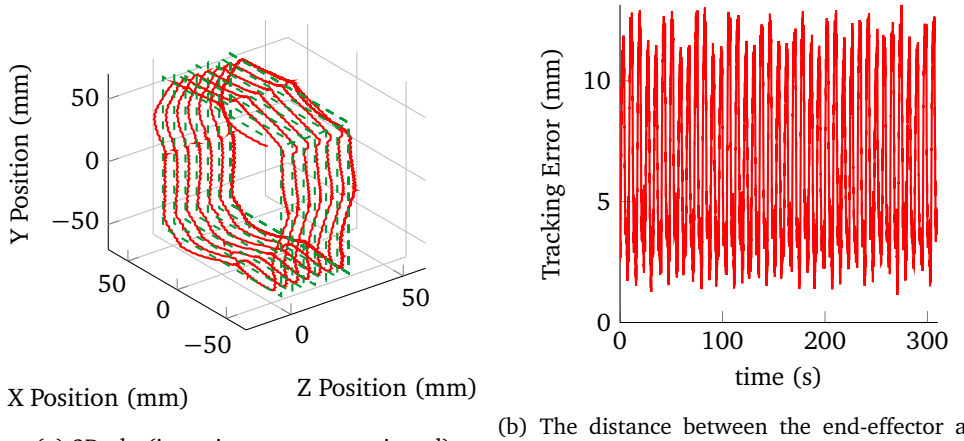


(a) 3D plot (input in dashed green, output in solid red)

(b) The distance between the end-effector and target

Figure 6.11: The closed-loop response to a cylinder shaped input

The beam shaped input has non-continuities at the corners of the squares. The closed-loop system performed better than the open-loop system. The squares were less pillow shaped than in the open-loop case. The highest inaccuracy occurred at the corners of the square. The average tracking error was 7.2 mm with a deviation of 1.5 mm and a maximum tracking error of 10.7 mm.

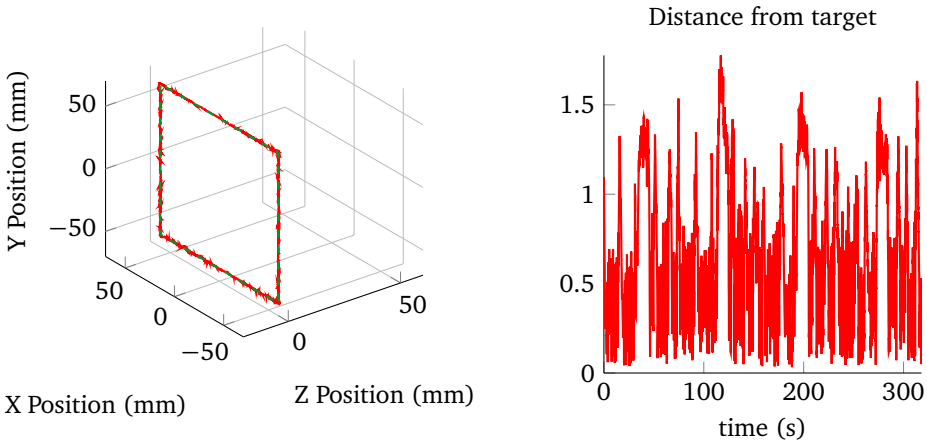


(a) 3D plot(input in green, output in red)

(b) The distance between the end-effector and target

Figure 6.12: The closed-loop response to a beam shaped input

All these shapes were performed ~50% beyond the required speed of the device. The device could also be operated at a lower speed. In figure 6.13 the response when the device was ran at 0.15 mm/s can be seen. As can be seen in this figure the system was capable of following a square shape more accurately at a lower speed. The average tracking error in this case was 0.7 mm with a deviation of 0.4 mm and a maximum error of 1.8 mm. This is due to the time delay feedback problem being less pronounced.



(a) 3D plot(input in dashed green, output in solid red)

(b) The distance between the end-effector and target

Figure 6.13: The closed-loop response to a slow square shaped input

6.3 Results

Table 6.1 shows that the closed loop performance at 6 seconds was better than the open loop performance. The accuracy (at 6 seconds) was increased. The precision however was lower because of the sensitivity of the feedback loop to measurement noise. The overshoot is less than 4%.

Input	Open Loop		Closed Loop		
	Accuracy (mm)	Precision (mm)	Accuracy (mm)	Precision (mm)	Overshoot (mm)
Step X	14.5	0.1	2.1	0.2	2.2
Step Y	15.8	0.1	1.4	0.2	2.2
Step X-Y	21.0	0.3	2.0	0.5	2.2

Table 6.1: Step response results

As mentioned before the performance increase when performing geometrical shapes was less than increase with the step responses. The tracking errors for the closed and open-loop system are quantified in table 6.2. The same pattern as with the step responses is visible here, the average and maximum tracking error go down and the precision is worse. The performance could be increased by slowing it down as can be seen by the sub-millimetre average tracking error for the slow square pattern. The open-loop model did not improve by slowing the system down.

Input	Open Loop			Closed Loop		
	Avg (mm)	σ (mm)	Max (mm)	Avg (mm)	σ (mm)	Max (mm)
Circle 15 degree/s	17.6	1.8	21.3	7.4	2.6	12.8
Cylinder 15 degree/s	17.9	1.7	21.4	7.2	2.5	12.3
Beam 15 mm/s	19.8	2.7	26.3	5.6	1.5	10.7
Square 0.15 mm/s	-	-	-	0.7	0.4	1.8

Table 6.2: Shape tracking errors

6.4 Human Input

The Omni Phantom was also used as an input device. A simple cross shape pattern was printed and a person tried to follow this pattern as precisely as possible using the Omni, this resulted in input speeds of <10 mm/s. If the person controlling the phantom moved faster the tracking error increased. The slower response had the added advantage of filtering out small involuntary movements. The Phantom Omni setup can be seen in figure 6.14. This process was repeated 5 times.

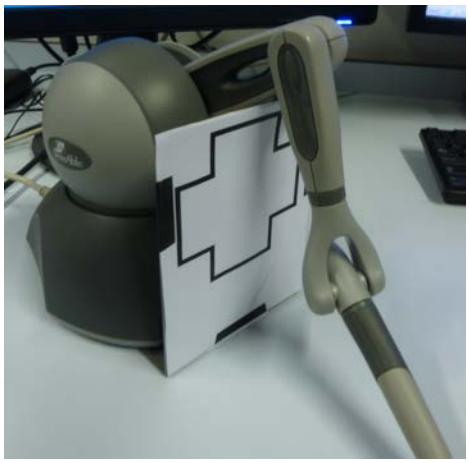
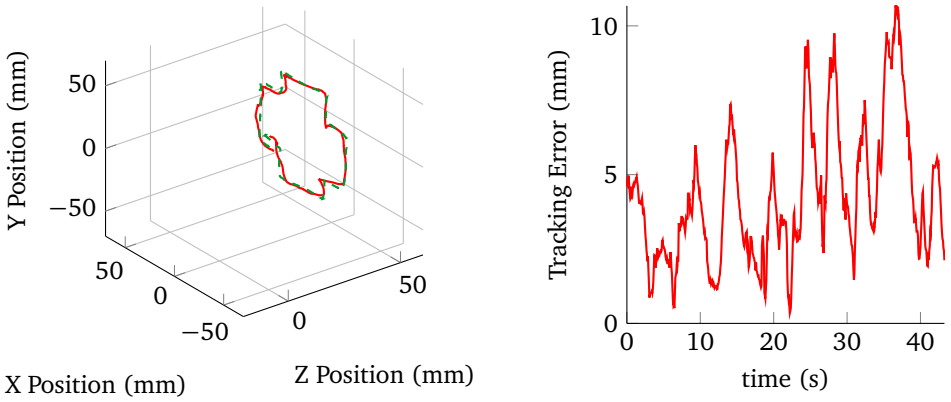


Figure 6.14: The Phantom Omni with the cross pattern

The response of one of these experiments is plotted in figure 6.15. The robot was capable of following the input pattern. The highest tracking errors occurred at the corners of the cross shape.



(a) 3D plot(input in dashed green, output in solid red)

(b) The distance between the end-effector and target

Figure 6.15: The response of the system to the human input

The average tracking error was 4.4 mm with a standard deviation of 2.3 mm while the maximum tracking error was 10 mm.

6.5 Smaller robot

The smaller tendon guide design was also experimented with to see whether the current design could be reduced in size. The device worked and the feedback system was capable of tracking the end effector. However, because of the shortcomings in the physical construction of the robot the experiments were inconclusive.

The tendons stretched too much. This meant that the controller started to compensate to get the end-effector closer to the desired position. It did this by increasing the desired deflection and then recalculating the tendon length required for that deflection. But the tendon stretched so much that the controller outputted a X or Y that was so large that equation (2.19) could not be solved. The resulting behaviour can be seen in figure 6.16, in this experiment the gains were turned down to check the behaviour of the robot. A step in the X-direction was applied. It continued to work for a while but in the end the desired position that the controller sends to the steering thread increased too much, resulting in instability.

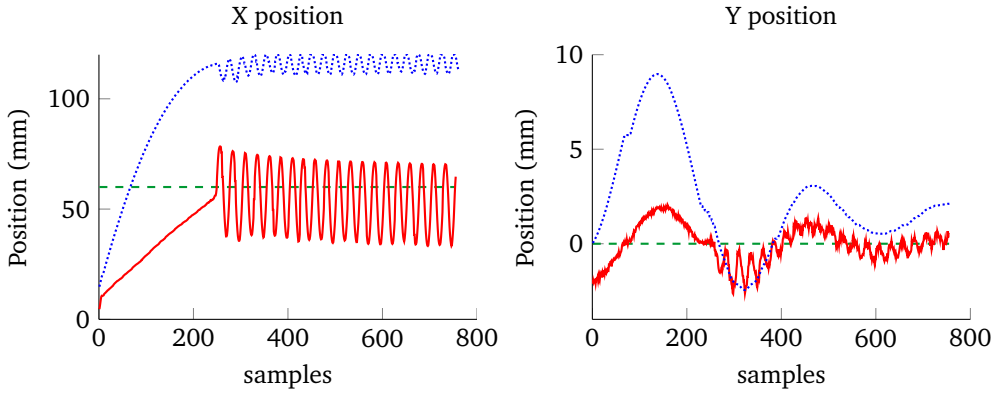


Figure 6.16: The Closed-loop Step response of the smaller continuum robot (input in dashed green, measurement in solid red and controller output in dotted blue)

Also, the friction between the tendons and the tendon guides was too much. This resulted in a jittery motion if faster movement was attempted. This was because the tendons stretched unequally between the segments and then settled when the force got too high. This settling also sometimes happened after relatively long periods of time (>1 sec). An attempt was made to pre-tension the tendons further to pre-stretch the tendons. But this resulted in failure of the robot with glue connection failing at the base of the robot.

6.6 Discussion

The closed-loop system performance was better than that of the open-loop system. There was an increase in accuracy when the feedback system was used. The effect of friction and fabrication imperfections were reduced. However, the performance gain was limited because of the time delay in the feedback system. This was especially apparent when using shapes (constantly changing inputs). Therefore the idea of combining a simple kinematic model with a 3D position feedback controller is most likely sound but the implementation can be improved.

The requirements as set out in section 3.1 were met. The ≥10 mm/s speed was attained. The motors had torque in reserve and could therefore turn faster than their nominal speed. They also accelerated faster to their maximum speed than was assumed in the design of the actuator. The device could also bend at 45°. The closed loop system could deflect to 60 mm repeatedly.

The setup was capable of following human input as long as the person controlling the device makes did not make very sudden input changes. This meant that small involuntary movements of the controlling person were filtered out.

Miniaturization is not possible without improving the construction of the device. The actuator and feedback combination work with the smaller robot. But the friction needs to be reduced and the tendons need to be more rigid.

In the next chapter this study is concluded and some recommendations are given.

Chapter 7

Conclusions

This chapter presents the conclusions in section 7.1 and the recommendations in 7.2.

7.1 Conclusion

This study tried to show that a simple kinematic model is capable of driving a continuum robot when combined with a 3D position feedback controller. The kinematic model as described in chapter 2 model worked. The actuator was capable of driving the continuum robot within the requirements that were set out in chapter 3. The controller improved the performance of a 160 mm length, 10 mm radius tendon guide robot. While the open-loop setup had an accuracy of $\sim$15 mm during a maximum deflection step response this was improved to 2 mm with the feedback system. While the open loop setup tracked a beam shape at maximum speed with an average tracking error of 20 mm, the closed-loop setup did so with an average tracking error of 5.6 mm.

There were however some limitations. Due to the time delay in the feedback system the maximum performance envelope could not be established. It is likely that the performance could be improved when a feedback signal is applied with less time delay. The construction meant that the miniaturisation was not possible. Disturbances such as friction became too large and the system became unstable.

7.2 Recommendations

Because of the things learned during this study recommendations for improved design and further research can be made. Section 7.2.1, discusses some ways in which the current design can be improved. Section 7.2.2 gives some ideas for further research.

7.2.1 Design Improvements

There are some points of the design that could be changed to improve the performance of the setup.

Tendon guides

The tendons guides were 3D printed objects fixed to the nitinol backbone. The super glue was not an ideal method of fixing the backbone, the 5 mm radius continuum robot failed. A specific type of glue that is more suited for glueing Nitinol and the teflon based 3D printed material together may improve the connection.

The tendon guides could also be made from metal. This is more sturdy and therefore they could be thinner. This would result in less friction and more similarity with the constant curvature model. These metal cable guides could be soldered onto the backbone. Soldering nitinol is difficult but it can be done with the right flux and the right kind of solder [17].

Tendons

The tendons were made out of fishing line. This should be changed to a better material (stronger, more rigid). Like, for example, titanium medical wire.

Feedback system

The feedback system had a delay. This delay limits the performance of the closed-loop system. The setup could be improved by using a custom grabber card and cameras increasing the framerate and reducing the delay. A switch to a different operating system such as real-time Linux may also lead to better results.

7.2.2 Further Research

Besides improvements to the current design there are also some new avenues of research that can be pursued to improve the performance of the continuum robot.

Improved Controller

The controller can be improved in multiple ways. The controller worked by feeding the kinematic model an desired position that differs from the actual desired position, therefore compensating for the error. This meant that there is a limit to the maximum input the controller could deliver. To be exact, the maximum input is that of a 90° bend because at that point the deflection is at a theoretical maximum. The controller could perhaps work at a tendon length level or perhaps change to velocity control.

Also, a different type of controllers could be used. The system is non-linear and a PID-controller is a linear controller. This controller is only stable because the gains are tuned in such a way that there is a large phase and gain margin. A non-linear controller may be better suited for the system, such as a sliding mode controller.

Different Image Modality

An optical feedback system is obviously of limited use *in vivo*, therefore an imaging modality should be considered that is usable inside the human body. Ultrasound is a possibility, it has been proven capable of tracking these kinds of robots [15].

FBG sensors

Another possibility that connects to other research being done at the medical robotics group at the University of Twente is using Fibre Bragg Grating (FBG) sensors in the robot's backbone. If these sensors are integrated into the backbone of the robot they would be capable of measuring strain inside the backbone. These strains could be used to calculate the curvature of the backbone.

Appendix A

Camera Calibration

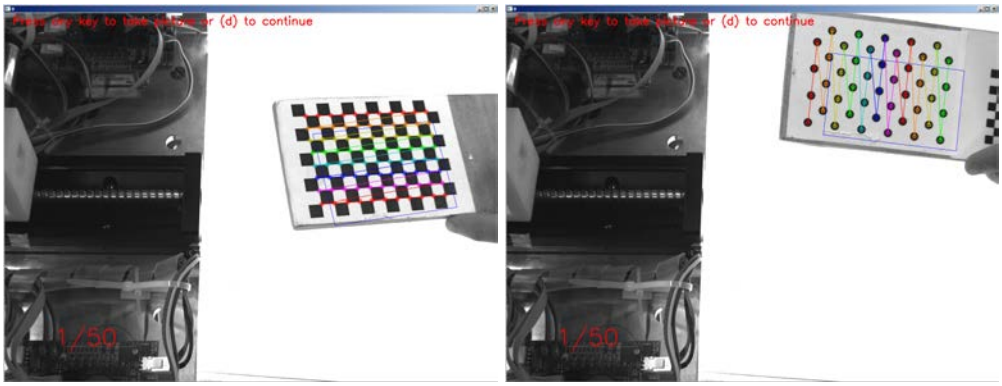
The systems used a stereoscopic camera setup as a feedback system as discussed in section 4.1. For this setup to be useful the cameras first needed to be calibrated. To do this a program was created. This program first calibrated the cameras by a process discussed in section A.1.1. The relation between the two cameras was also established, this process is described in section A.1.2. These two matrices were used in the triangulation algorithm as discussed in section 5.1.3. For the image processing to function some variables have to be tweaked this is also done by this program as described in section A.2. This program was a C++ win32 application using functions provided by the OpenCV library.

A.1 Calibration

The main goal of calibration is getting the camera matrix. This gives the relation between a point in space and a pixel on the camera sensor. There are several different methods of doing this, such as using a known pattern to derive this matrix or using the vanishing points [3].

A.1.1 Camera Matrix

This program used regular patterns. The OpenCV library has functions for finding a checkerboard pattern or circle pattern in an image. This program supported both patterns as can be seen in figure A.1. The patterns were printed using a laser printer (HP Laserjet 4050) and carefully glued to a flat surface. They were examined under a microscope and using a 2D linear stage with a linear encoder (Miyoto AT112) to check how precise the pattern was. The pattern was printed with an accuracy of 50 μm, sufficient for this process.



(a) A checkerboard pattern (b) A circle pattern

Figure A.1: Calibration patterns

The program can take a continuous stream of images and automatically find the checkboard pattern and store the image points. It can also take an image every time the user presses a button. The button approach had the best results because the user can hold the pattern completely still. After each hit a square was drawn on the screen to show the user where a pattern had already been found, the user should try to cover the entire image for the best results. After a set number of patterns had been found the program started calculating the camera matrix. The OpenCV library does this by estimating the camera matrix by running a Levenberg-Marquardt optimization algorithm to minimize the reprojection error over all the patterns. That means that for all the images the algorithm estimates the pose of the pattern, the rotation and translation of the pattern in relation to the camera. The reprojection error is the distance between the coordinates of the points in the image that was taken and the coordinates of the points in the image when they are reprojected using the camera matrix from their estimated pose. The program rejected the patterns with reprojection errors of twice the average and then repeated the process. This was done to reject faulty measurements.

A.1.2 Stereo

The stereo calibration repeats the pattern detection process of the normal calibration. Only this time images are taken from both cameras as can be seen in figure A.2.

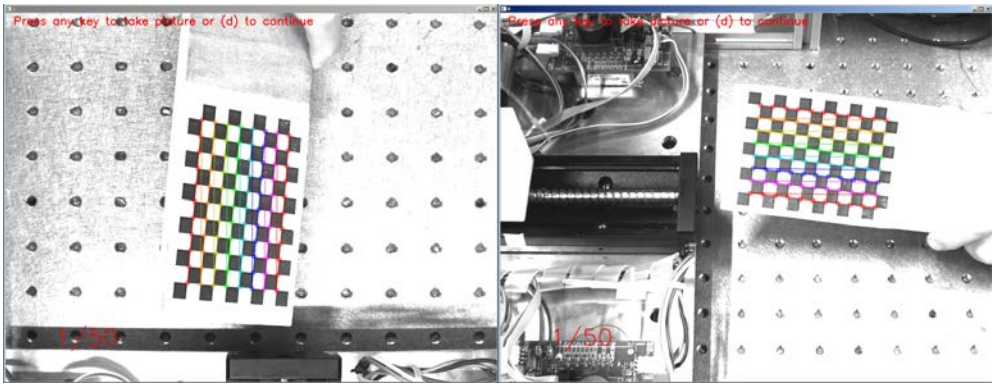


Figure A.2: Stereo calibration

In this case the pattern should be asymmetrical so that the algorithm can correctly establish the pose. The OpenCV algorithm takes the pose of the pattern in relation to the first camera (C_1, R_1) and the pose of the pattern in relation to the second camera (C_2, R_2) and tries to solve equation (A.1) to get R, the rotation from camera 1 to camera 2 and T the translation between the two.

$$R_2 = R \star R_1 T_1 = R \star T_1 + T \tag{A.1}$$

A.2 Setup

The software needed to know where the origin of the system (figure 2.2) was. The user needed to tell the software where this point is in the two camera images. For this a fixed marked point on the linear stage was used as can be seen in figure A.5. This point was then triangulated in the 3D space. This point was translated $(28.0, -35.7, 0.0)$ mm in relation to the origin.

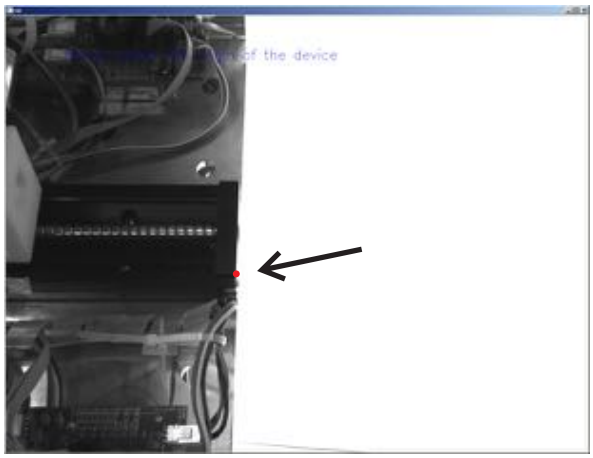


Figure A.3: Setting the origin

To speed up and simplify the image processing part of the software only part of the image was processed. So the user needed to set this region of interest (ROI). This was also done by the camera calibration program as can be seen in figure A.4

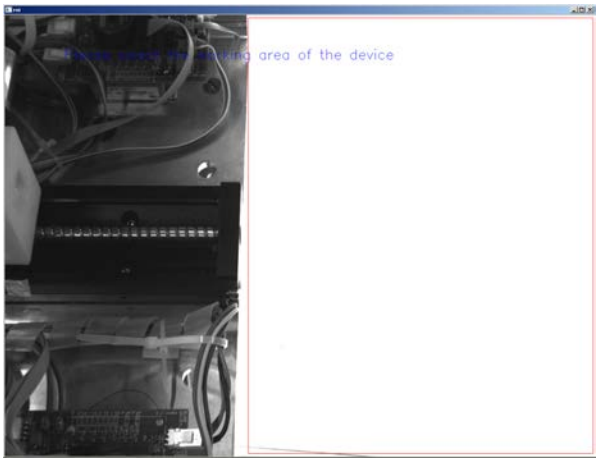


Figure A.4: Setting the roi

The image processing part of the software (see section 5.1.2) needed to be tuned. This could also be done with the camera calibration program as well. It offered three options to tweak as can be seen in figure A.5. The threshold, which determined the greyscale level at which something is detected as being part of the robot. The erosion level in pixels, which got rid of small speckles of noise and the mass rejection. Everything with a pixel mass below this level was rejected.

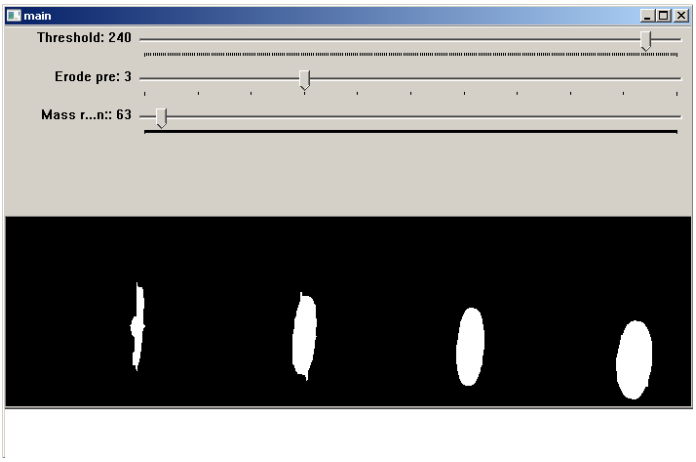


Figure A.5: Tuning the image processing algorithm

Appendix B

Parts List

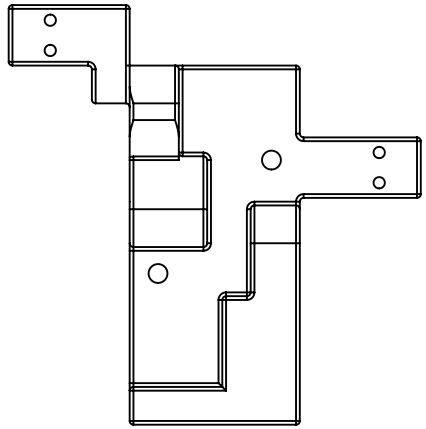
Part	Supplier	Description	Part number	URL
Single Axis Actuator LX30	Misumi	Linear Stage of the actuator	LX3005-B1-N-300	http://sg.misumi-ec.com/asia/ItemDetail/10300075450.html
Couplings - Sleeved	Misumi	Coupling between the motor and the linear stage	CPF16-4-6	http://th.misumi-ec.com/asia/ItemDetail/10300125420.html
Cable Carriers -Slit-	Misumi	Cable Carriers for the cables to the motors on the carriage	SE045F-16-018-20	http://in.misumi-ec.com/asia/ItemDetail/10300073350.html
EC-max 22 ⌀22 mm, brushless, 12 Watt	Maxon	The Motors	283840	http://www.maxonmotor.nl/maxon/view/product/motor/ecmotor/ecmax/ecmax22/283840
Encoder MR, Type ML, 512 CPT, 3 Channels, with Line Driver	Maxon	Encoders of the motors	201940	http://www.maxonmotor.nl/maxon/view/product/sensor/encoder/imp512/201940
Planetary Gearhead GP 22 C ⌀22 mm, 0.5 - 2.0 Nm, Ceramic Version	Maxon	The gearbox of the tendon pulling motors	143988	http://www.maxonmotor.nl/maxon/view/product/gear/planetary/gp22/143988
Planetary Gearhead GP 22 C ⌀22 mm, 0.5 - 2.0 Nm, Ceramic Version	Maxon	The gearbox of the linear stage motor	143976	http://www.maxonmotor.nl/maxon/view/product/gear/planetary/gp22/143976
Whistle Controller 2.5/60	Elmo	The motor controllers		http://www.elmomc.com/products/whistle-digital-servo-drive-main.htm

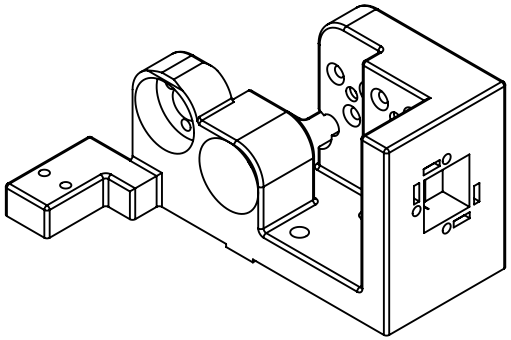
Table B.1: The Parts list

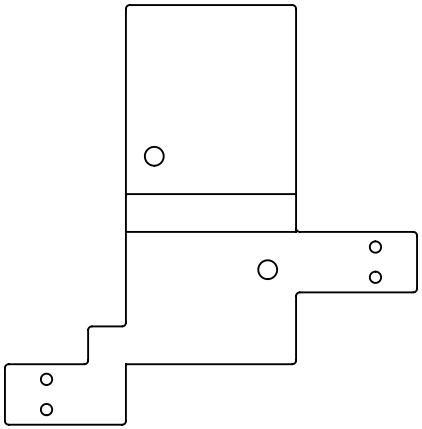
Appendix C

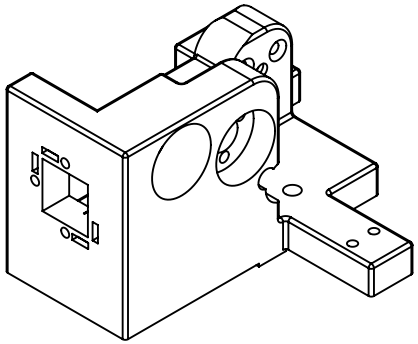
Cad Drawings

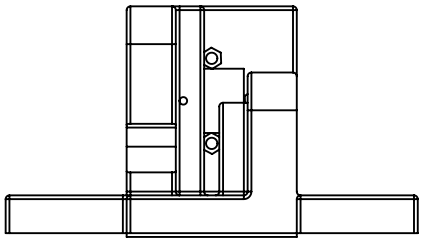
In this appendix the CAD-drawings of the 3D-printed and machined parts are shown.

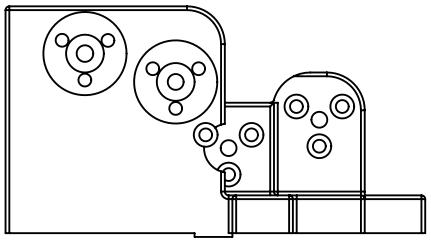


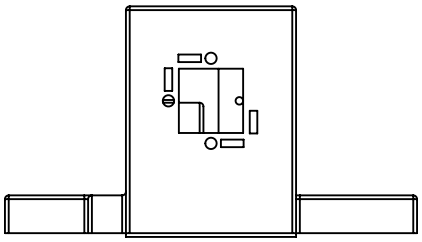


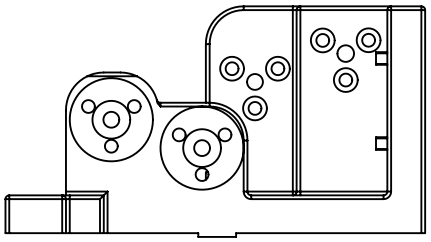




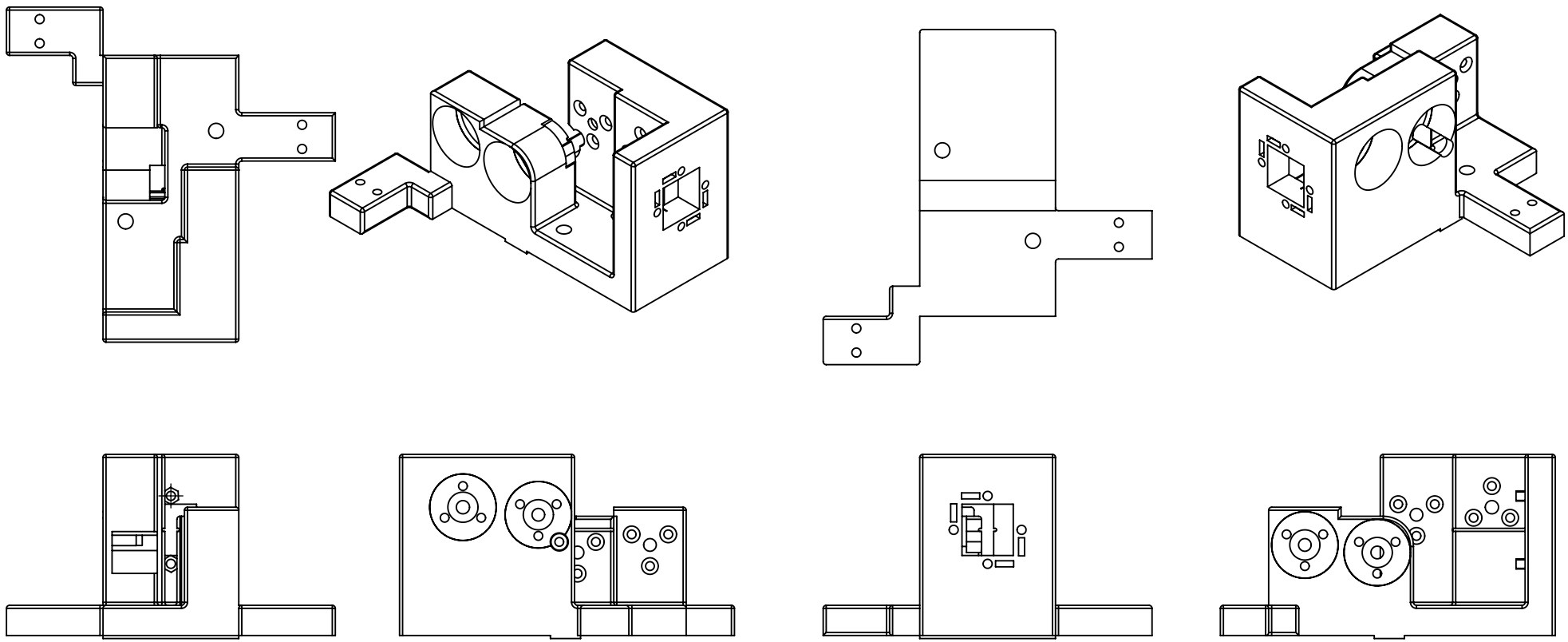




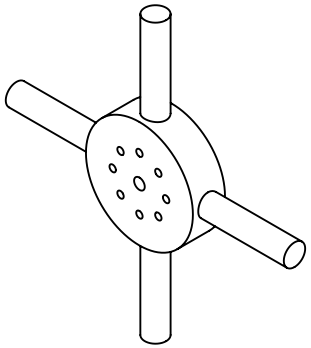


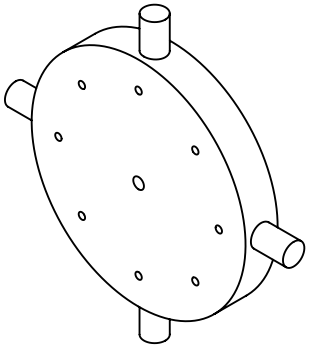


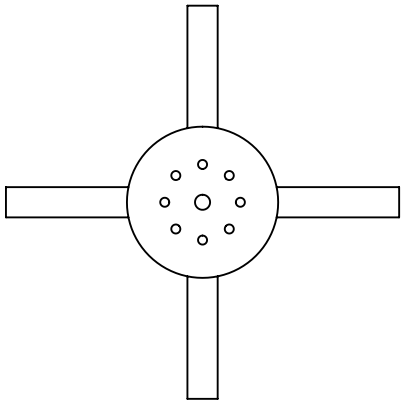
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION
	NAME	SIGNATURE	DATE				TITLE:
DRAWN							
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:			DWG NO. Carriage 10mm / A4
				WEIGHT:			SCALE:1:2 / SHEET 1 OF 1

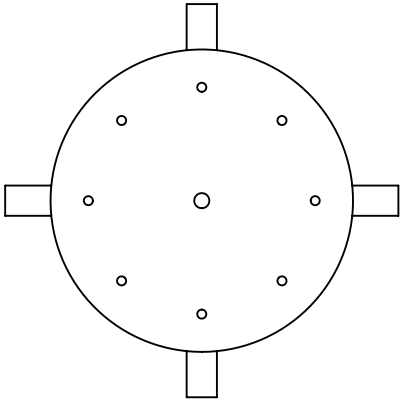


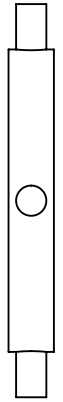
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE		TITLE:	
DRAWN						
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL:	DWG NO. Carriage 5mm	A4
				WEIGHT:	SCALE:1:2	SHEET 1 OF 1



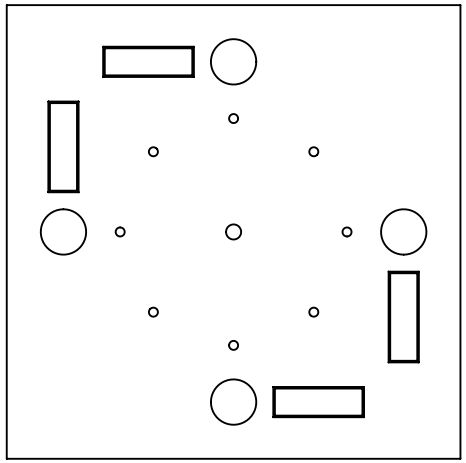


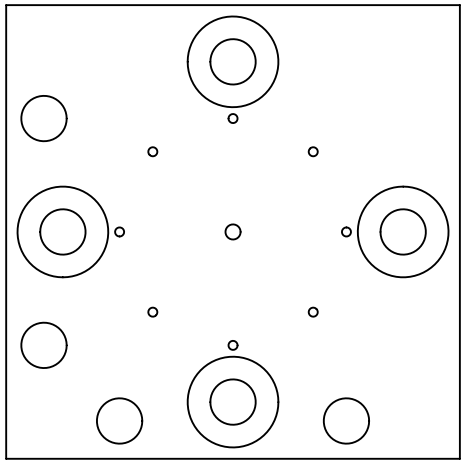


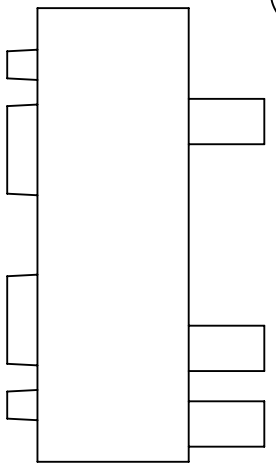


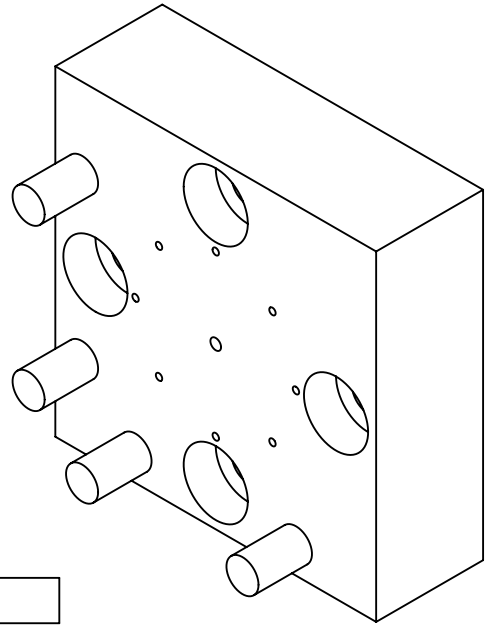


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION
	NAME	SIGNATURE	DATE				TITLE:
DRAWN							
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:			DWG NO. Tendon Guides / A4
				WEIGHT:			SCALE:2:1 / SHEET 1 OF 1

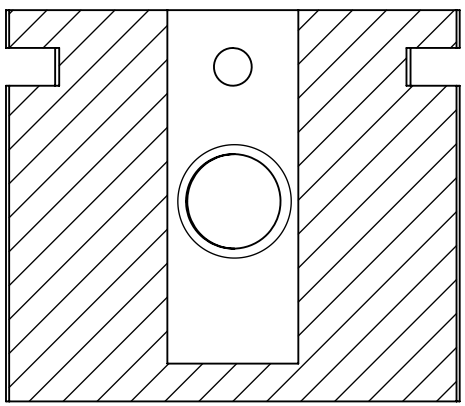


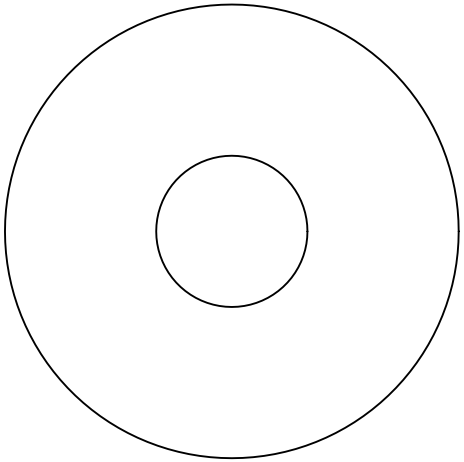


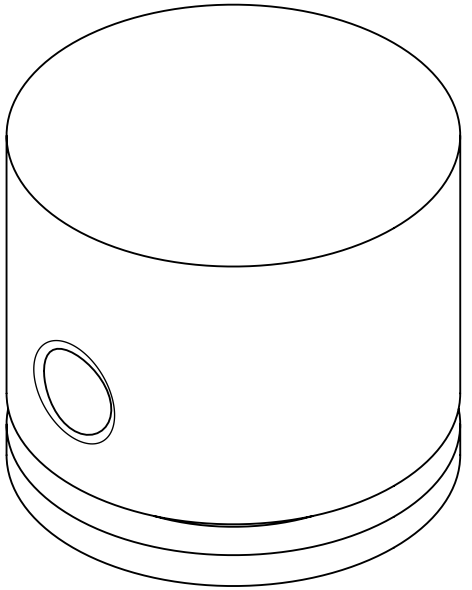


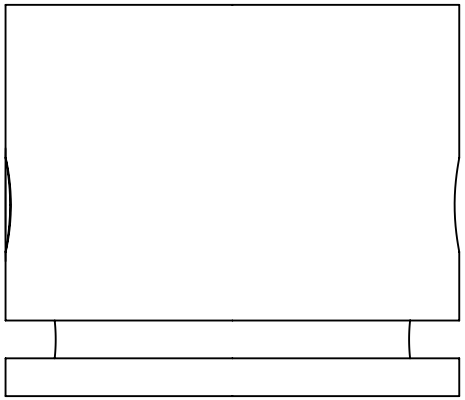


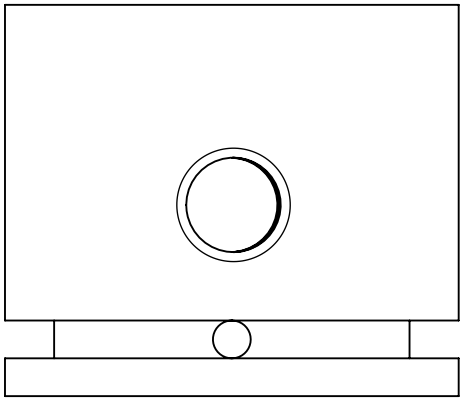
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:			DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION
	NAME	SIGNATURE	DATE			TITLE:
DRAWN						
CHK'D						
APPV'D						
MFG						
Q.A				MATERIAL:		DWG NO. Base Plates / A4
				WEIGHT:		SCALE:2:1 / SHEET 1 OF 1











UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:				DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION
	NAME	SIGNATURE	DATE				TITLE:
DRAWN							
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:			DWG NO. pulley / A4
				WEIGHT:			SCALE:5:1 / SHEET 1 OF 1

Bibliography

[1] A. Bajo, R. E. Goldman, and N. Simaan. Configuration and joint feedback for enhanced performance of multi-segment continuum robots. In *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pages 2905–2912, 2011.

[2] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury. Mechanics modeling of tendon-driven continuum manipulators. 24(6):1262–1273, 2008.

[3] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes, 1999.

[4] I. S. Godage, D. T. Branson, E. Guglielmino, G. A. Medrano-Cerda, and D. G. Caldwell. Shape function-based kinematics and dynamics for variable length continuum robotic arms. In *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pages 452–457, 2011.

[5] I. A. Gravagne, C. D. Rahn, and I. D. Walker. Large deflection dynamics and control for planar continuum robots. 8(2):299–307, 2003.

[6] M. W. Hannan and I. D. Walker. The 'elephant trunk' manipulator, design and implementation. In *Proc. IEEE/ASME Int Advanced Intelligent Mechatronics Conf*, volume 1, pages 14–19, 2001.

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in computer Vision*. Cambridge University Press, 2nd edition, 2003.

[8] Richard I. Hartley and Peter Sturm. Triangulation. *COMPUTER VISION AND IMAGE UNDERSTANDING*, 68(2):146–157, November 1997.

[9] J. Jayender, M. Azizian, and R. V. Patel. Autonomous image-guided robot-assisted active catheter insertion. 24(4):858–871, 2008.

[10] B. A. Jones and I. D. Walker. Kinematics for multisection continuum robots. 22(1):43–55, 2006.

[11] Aart J. Klijn, Marino Asselman, Marianne A W. Vijverberg, Pieter Dik, and Tom P V M. de Jong. The diameter of the rectum on ultrasonography as a diagnostic tool for constipation in children with dysfunctional voiding. *J Urol*, 172(5 Pt 1):1986–1988, Nov 2004.

[12] G. Z. Lum, S. K. Mustafa, H. R. Lim, W. B. Lim, G. Yang, and S. H. Yeo. Design and motion control of a cable-driven dexterous robotic arm. In *Proc. IEEE Conf. Sustainable Utilization and Development in Engineering and Technology (STUDENT)*, pages 106–111, 2010.

[13] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. D. Walker, B. A. Jones, M. Pritts, D. Dienno, M. Grissom, and C. D. Rahn. Field trials and testing of the octarm continuum manipulator. In *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2006*, pages 2336–2341, 2006.

[14] R. S. Penning, Jinwoo Jung, J. A. Borgstadt, N. J. Ferrier, and M. R. Zinn. Towards closed loop control of a continuum robotic manipulator for medical applications. In *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pages 4822–4827, 2011.

[15] Hongliang Ren, Nikolay V. Vasilyev, and Pierre E. Dupont. Detection of curved robots using 3d ultrasound. In *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, pages 2083–2089, 2011.

[16] G. Robinson and J. B. C. Davies. Continuum robots - a state of the art. In *Proc. IEEE Int Robotics and Automation Conf*, volume 4, pages 2849–2854, 1999.

[17] S.M. Russell and A.R. Pelton. *SMST-2000: Proceedings of the International Conference on Shape Memory and Superelastic Technologies, 30 April to 4 May 2000, Asilomar Conference Center, Pacific Grove, California, USA*. SMST, The International Organization on Shape Memory and Superelastic Technology, 2001.

[18] Intuitive Surgical. Intuitive surgical. http://www.davincisurgery.com/.

[19] R. J. Webster and B. A. Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.

[20] Kai Xu, R. E. Goldman, Jienan Ding, P. K. Allen, D. L. Fowler, and N. Simaan. System design of an insertable robotic effector platform for single port access (spa) surgery. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2009*, pages 5546–5552, 2009.

[21] Song-Ling Yan and Mark Thompson-Fawcett. Notes: new dimension of minimally invasive surgery. *ANZ J Surg*, 79(5):337–343, May 2009.