MASTER THESIS

Sensorization and Control of a Multi-Segment Tendon Continuum Robot for Endoscopic Procedures

Iven van Horck

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF BIOMECHANICAL ENGINEERING

EXAMINATION COMMITTEE:
prof.dr.ir. H. van der Kooij
dr. A. Sadeghi
ir. N. Willemstein
dr.ir. M. Vlutters
dr. I. Tamadon

UNIVERSITY OF TWENTE.

Soft Robotics Lab

April 18, 2025

Contents

Acknowledgements

I would like to thank my supervisors, in particular, dr. Ali Sadeghi, ir. Nick Willemstein and dr. Mark Vlutters, for their valuable insights and assistance during the supervision of my thesis. I would also like to thank my girlfriend, my lab associates, and my friends for helping me whenever I needed it and for making my thesis period such an enjoyable experience.

Sensorization and Control of a Multi-Segment Tendon Continuum Robot for Endoscopic Procedures

Iven van Horck

April 18, 2025

Abstract

Endoscopy is a common medical procedure, performed millions of times per year worldwide, to diagnose and treat gastro-intestinal and respiratory diseases. These procedures are, however, not without risk. There is a chance that improper control of the endoscope by the endoscopist, caused by a lack of information about the robot's surroundings, can cause the endoscope to perforate the walls of an organ, which is life-threatening. This thesis presents an 8 DOF tendon continuum robot, that is capable of avoiding the walls of the organ its passing through. It has 24 optical reflective sensors on the robot's sides, pointed outwards in four sets of six. These sense the robot's surroundings up to a distance of 3.5 cm, with a field of view of 40.8° per sensor. Two control methods evaluated: 1) A PID controller that controlled the 8 DOF as 4 sets of 2 DOF. 2) A model-based (MB) controller that uses the constant curvature model to control all 8 DOF. In simulations, the MB controller had a $92.6\%(p = 0.000)$ loss reduction compared to the PID controller when positioning the robot in a desired location. Physical robot experiments were performed whereby an obstacle was moved towards the robot to measure the robot's deflection. If this obstacle was visible only to the fourth segment from the base, then the the MB controller was able to avoid the obstacle for a 92.9% $(p = 1.0 \cdot 10^{-10})$ longer distance than the PID controller. If the obstacle was visible to both the fourth and third segment, the MB controller outperformed the PID controller by 65.5% $(p = 1.8 \cdot 10^{-5})$. If the obstacle was visible only to the third segment, the MB controller is outperformed by the PID controller by 23.5% $(p = 1.9 \cdot 10^{-6})$. Overall, the robot and the MB controller demonstrated effective obstacle avoidance and adaptability within constrained environments. Further research should focus on reducing the robot size, to more closely align its design with endoscopes.

1 Introduction

One of the main methods that allows physicians to visually inspect the internal organs of their patients is endoscopy. It is a common medical procedure, performed over 22.2 million times in 2019 in the US alone [1]. These procedures are vital steps in the diagnosis of illnesses of a patient, either by visually inspecting an abnormality present on organ tissue or by taking samples for histopathology[2]. The latter of these procedures can only be done with flexible endoscopes.
Flexible endoscopes are one of the two main types of endoscopes, the other being rigid endoscopes[3]. Rigid endoscopes are commonly used in minimally invasive procedures. Flexible endoscopes are instead used for a wider range of diagnostic and therapeutic applications. It can perform these procedures whilst inside various cavities of the human body, also called the lumen[4], such as the GastroIntestinal (GI) or the respiratory system[3].
Endoscopes are made up of different parts. There is the handle, which the endoscopist uses to control the endoscope. There is also the insertion tube, which is the part of the endoscope that enters the human body. The flexible tip of the endoscope is connected to the insertion tube. On the tip, there is a camera, as well as a channel to allow the usage of tools during the procedure. The flexible tips curvature can be actively controlled by the physician. This actuation is done with a set of tendons, that are linked to a wheel or lever on the handle of the endoscope. in essence, this is a manually controlled Tendon Continuum Robot (TDCR) [5]. The curvature control is necessary, as this is used to steer the endoscope as it moves through the body during its insertion. It is also used to orient the tip towards a region of interest inside the lumen during the procedure.

Controlling the endoscopes is something that requires experience, as its controls are not intuitive [6]. There is a strong correlation between the speed of the endoscope insertion and the experience level of an endoscopist, showing that learning to control an endoscope takes time and experience to do successfully.

Improper control of the endoscope can cause complications to occur. The tip might hit the walls of the lumen, or shear against them during the procedure [7], causing the wall of the lumen to be perforated. This perforation can also occur further down along the endoscope, along the bent section of an endoscope. As the endoscope is manoeuvred, this section of the tube might start to put pressure against the walls of the lumen, and if sufficient pressure is applied, a perforation will also occur at this location [8].

Perforation of the walls of the lumen can be life-threatening. If they occur in the upper GI tract have a mortality rate of 2% and 36% [9], and can occur between 1 in 1100 and 1 in 2500 cases. In colonoscopies, perforations occur between 0.03% and 0.8% of cases.

The risk of a perforation can be reduced by better control of the endoscope. More comprehensive information about the surroundings of the endoscope can help in that control. One of the ways control was improved was by using haptic devices [6]. These enabled the user to intuitively steer the tip of the endoscope, and enable a more intuitive method of control. This did not fully solve the issue, however, as the endoscopists were more adept at using the conventional control method.

Other research looked into the automation of the endoscope guidance and control. One method is to use the front view of the endoscope camera to automate. [10]. This method limits can perforations of the tip of the endoscope by trying to keep the endoscope view in the centre of the lumen. This is effective, however, it only works for obstacles that can be seen through the front-facing camera. Perforations caused by the sides of the flexible tip section or the insertion tube are not prevented. This method also only works on endoscopes that have a front-facing camera. Not all endoscopes contain a front-facing camera, such as echo-scopes [3] and as such this method might not work. Looking more broadly in the field of continuum robotics, a lot of research is focused on automated control of TDCR using various control methods[11]. Many of these works use either an external camera setup to track and estimate the pose of the robot, or another method of externally measuring the pose of the robot to calculate the controller response. In a medical context, this can be done using an external imaging method such as ultrasound [12]. However, this adds another expensive medical device to control the first, which is undesired. Some other methods [13] use force feedback from the endoscopes surroundings for automated control through that environment. This, however, means that the robot must already be in contact with the walls of the lumen before it can start avoiding them. Placing sensors upon the robot itself is a common method to inform controllers [14], however it is only rarely applied in, TDCR context. A TDCR [15] was developed that did use optical and force sensors along the length of the robot to estimate the robot surroundings. Its applicability to endoscopy is limited however, as the sensor suite that was too large to implement in endoscopes. Furthermore, the sensor refresh rate was low, which complicated control.

There are methods that are capable of measuring the surroundings of an endoscope. Capacitive sensors [16] can detect the distance of the tip from its surrounding environment. Echo-endoscopes and optical coherence tomography endoscope sensors are also not pointed in the forward direction[3], although these are used in a diagnostic capacity, rather than to improve control.

Another advancement in endoscopy itself is the creation of more complex multi-bending endoscopes [17]. These endoscopes have an additional bending segment, doubling the Degrees of Freedom (DOF) of the continuum robot. Their design allows the user to more effectively reach regions of interest in the lumen that other endoscopes cannot. However, The control of these endoscopes is more difficult than conventional endoscopes [18],

To generalize, it can be seen that a lot of work has been done to make endoscopy a safer procedure for the patient and in automating the the control of tendon continuum robots. However, these control methods often rely on external sensing methods, or only use the front facing camera of the endoscope. This all means that the problem of perforations of the walls of the lumen has not yet been solved by the field, and it shows that there is reason to explore a solution to these problems. From this, the question can be posed:

How can a multi-segment tendon continuum robot be designed and controlled so that it avoids the walls of the lumen?
Here, a novel fusion of endoscopy sensing and control will be utilized to both estimate the position of the robot segments within the lumen by using integrated sensors. It can then calculate a control response to ensure it will stay within the centre of the lumen.

2 Materials and Methods

The environment around which the robot is designed is the "lumen", which is defined as "the space inside a tube, blood-vessel or hollow organ" [4]. For example, he stomach, oesophagus, colon, and small intestines are all lumena.

These areas are highly diverse in their geometry. The stomach is an open cavity, the oesophagus is an approximately linear tube, whilst the colon and small intestines contain multitudes of curves in three-dimensional space [4]. The cross-sectional diameters of these lumena vary, with the stomach and large intestine being the largest at about 7 cm [4], while the small intestine and oesophagus are smaller. For this reason, the largest endoscopes have a diameter of $12.8\ mm$ [19]. As each endoscope has design requirements laid out by the specific geometry of its environment it will be used in, it will be impossible to create a single robot arm that is capable of performing well in all these environments. Instead, it was chosen to design the robot arm in such a way that it is capable of making omnidirectional curved bending motions in three-dimensional space to allow the testing and validation of sensorization and control. Further research can then focus on applying the concept to specific endoscope types. Regardless of the specific endoscope type, all designs must operate inside the human body. As such, the robot must be able to detect the distance to the walls of the lumen.

Endoscopic operation time varies, full endoscopic procedures in the oesophagus can last for 5.8 minutes [20], whilst the insertion time for some colonoscopies takes 13.5 minutes [21]. In all cases, however, this is so short that the robot must be able to react to its surroundings in a short time frame. So, real-time control [22] is required to ensure the functionality of this robot. Normal endoscope cameras have frame rates of 25 to 30 Hz [23], which is sufficient for the endoscopist to safely guide the endoscope through the human body. This means that the controller and sensor must also be able to have a control speed of at least 25 Hz. This controller must be designed so that it can avoids obstacles, and by doing so, conform the robot to its environment.

The requirements can be summarized as follows:

1. The robot must be able to sense the distance from itself to the walls of the lumen.
2. The robot arm must be smaller than $12.8\ mm$ in diameter.
3. The robot must be able to curve omnidirectionally in three-dimensional space.
4. The robot must be controllable in real-time, with a control loop speed of at least 25 Hz.
5. The robot must be able to avoid obstacles and conform itself to an environment.

2.1 Robot Design

The full robot that was designed can be seen in figure 1.D. The robot consists of a robot arm connected to a rectangular box containing servo actuators, with the control electronics placed on top.

The robot arm is analogous to the insertion tube of an endoscope, without its outer shell, and with modifications to simplify the construction. The sensors or the robot are spaced equally around its radius, upon 4 disks. The rest of the robot is analogous to the handle of the robot, which is responsible for the actuation of the arm in a specific configuration. Signal processing, controller algorithms and tendon actuation are all performed in this part. To establish this design, the general shape must first be ascertained. Specifically, the number of segments that can be actuated by the robot. For this, two approaches can be followed:

- An endoscope with only one bendable section and an otherwise passive insertion tube.
- An endoscope whose entire insertion tube consists of bendable segments, making it fully actuated.

The single bendable segment is the most similar to the current form factor of the majority of endoscopes. However, this design is limited in its range of motion, as it has, at most, 2 DOF. The development of Multi-bending endoscopes [17] shows that

an increase in the range of motion is desired. A fully actuated insertion tube has an increased DOF, but the construction is more complex, due to the increased amount of components. For example, the "Harvard Apparatus Flexible Endoscope" (HAFE) [24] has a length of 300 mm, with a bendable section with a length of 30 mm. Therefore, a fully actuated insertion tube would require 10 bendable segments, which gives 20 DOF, at the cost of a tenfold increase of tendons in the system.

In the end, it was chosen to construct the robot with the entire insertion tube consisting of bendable sections. The reasoning was mostly practical, as it was easier to convert actuated segments to passive segments than to add new actuated segments to a passive tube. This allows for the evaluation of the control scheme in both configurations. Therefore, the robot was designed to have four bendable segments. This allows for L and S configurations of the robotic arm, whilst not having a design so complex that it could not have been made in a reasonable time frame. The robot arm was chosen to to have a length of 25 cm as this was the length of the available spring which was used as the spine of the robot. The diameter of this spring was 9 mm.

2.1.1 Tendons and Actuation

To be able to control a tendon continuum robot in three dimensions, a minimum of three tendons are needed [5]. Actuation of a single segment with four tendons is also possible and allows control of the curvature with only two actuators instead of three. However, more tendons increase the complexity of the robot arm design, whilst more motors only require more space in the base of the robot. Having a less complex robot arm is preferable to having a smaller robot base, and as such, a three-tendon configuration was chosen.

For the actuators, position-controlled servos were used. These servos have an internal PID controller, which moves them to any angle between 0° and 180°. The robot can control this angle by transmitting a PWM signal to each of these motors. As there are four segments, with three tendons each, a total of 12 of these servos are required. Each of these tendons has its guiding holes in the robot disks, as seen in figure 1.C.1.

For a single segment, the three tendons are spaced 120° from one another, as seen in equation 6. To then space out all four segments equally, each set of tendons will need to be rotated with 30° compared to the previous, which is also accounted for in 6 with the N_{seg} term. This ensures that each of the segments has the same orientation definition in three-dimensional space. The tendons are connected to the robot with a pulley. These ensure a larger Δl_n is possible for the same rotation of the servo.

2.2 Sensors

The sensors of the robot were placed upon the robot itself. This allows for low latency integration of the sensor response, as well as an occlusion-free view of the surroundings of the robot [14].

Both capacitive sensors [16] and optical Time Of Flight (TOF) sensors [15] are sensor methods that have previously been used as proximity sensors on continuum robots in this manner. These implementations both had their difficulties. Capacitive sensors are prone to parasitic capacitance [16] and are difficult to manufacture. TOF sensors need time to converge to a solution, which might slow down the overall speed of data acquisition, and thus limit the speed of the robotic control[15]. Instead, in this research, reflective optical sensors were chosen. These were chosen as they are commercially available in a small size[25], whilst being cheap, and because of their ease of use and implementation. Optical proximity sensors also use a diverging light source [26], which might more readily pick up obstacles when compared to the laser of a TOF.

2.2.1 Sensor Design

The design is as follows: an infrared LED placed next to a photo-diode, called a Light Receiving Diode (LRD) for clarity. If the LED is turned on, the LRD measures a signal only if an obstacle reflects the light from the LED back into the LRD [26]. The intensity of the reflected light L_{raw} is proportional to the distance of the obstacle as follows:

$$L_{raw} \propto \frac{1}{distance^4} \tag{1}$$

This equation is idealized, and the usage of these sensors in a real environment complicates this. Most notably, external light sources will influence the brightness that is measured by the sensor, which reduces the accuracy of 1. This is solved by first measuring the light intensity $L_{ambient}$ whilst

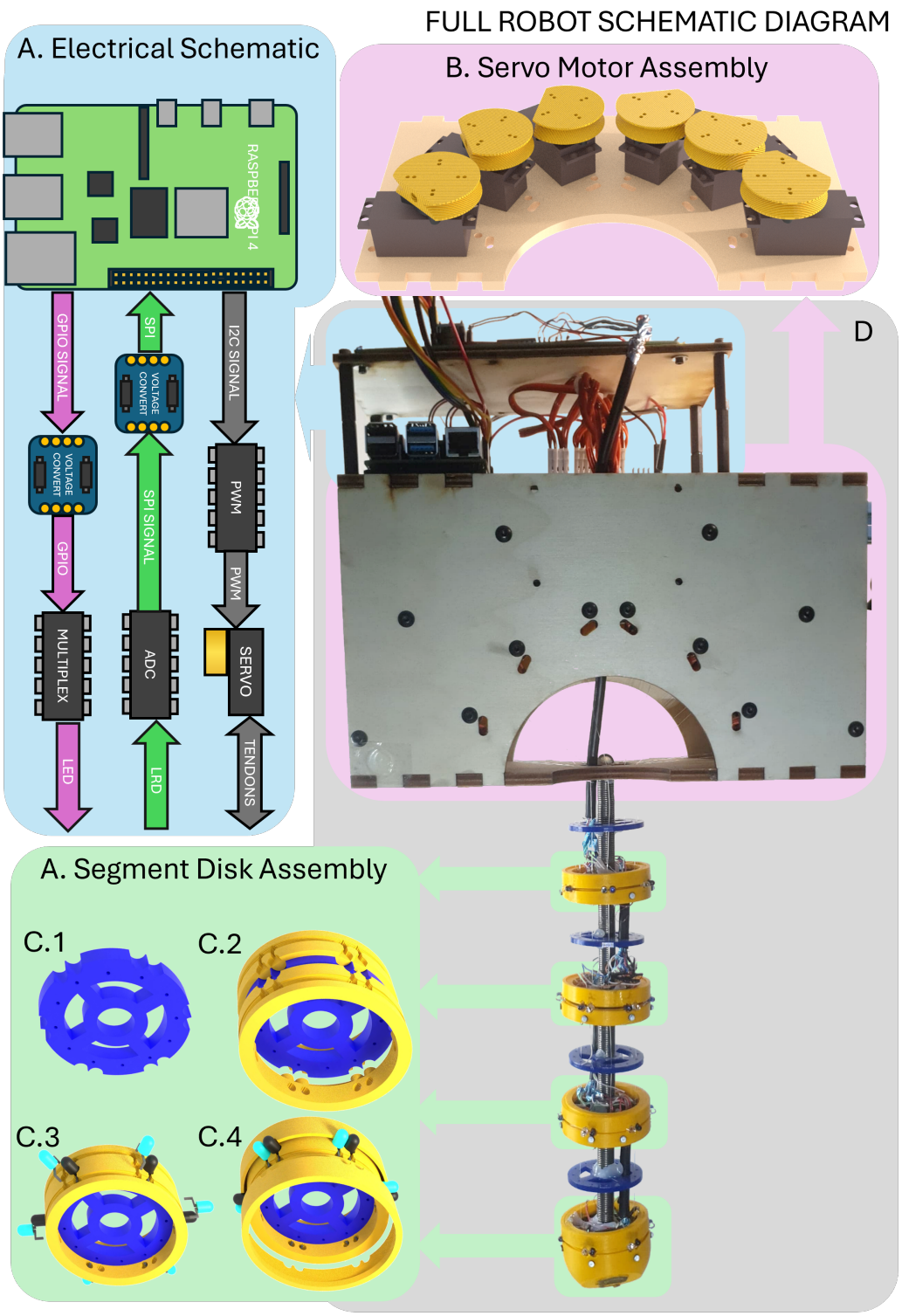

Figure 1: A shows the schematic diagram of the electronics, with the RPI, and the method to control the tendons (grey arrows) and the sensors (green and purple arrows). Power, ground, as well as auxiliary components are omitted for clarity. MULTIPLEX is the Analog Multiplexer chip. PWM is the Pulse Width Modulation multiplexer board. ADC is the Analog to Digital Converter chip. GPIO, SPI and I2C are three different communication protocols. The arrows indicate the direction of information flow. B shows the method by which the 12 motors are placed in the robot, with 2 of these plates. C shows the full assembly of the disks of the robot. C.1 shows the acrylic disk. C.2 shows how the two inner components are fitted onto the disk. C.3 shows how the sensors are fitted onto the inner components. C.4 shows how the outer components are fitted onto the inner components. D shows the full robot, with the components highlighted.

the LED is off, before measuring with the LED on. The "true" signal L_{signal} can then be calculated as:

$$L_{signal} = L_{raw} - L_{ambient} \tag{2}$$

To then be able to convert this signal into meaningful data, L_{signal} needs to be calibrated. This is done by measuring the maximum (L_{max}) and minimum (L_{min}) sensor response for each sensor and then applying min-max normalization [27]

$$L_n = \frac{L_{signal} - L_{min}}{L_{max} - L_{min}} \tag{3}$$

This results in $0 \leq L_n \leq 1$ for all sensors. However, L_n still follows the curve of 1. If this signal was used for control, it would bias the controller response, which is undesirable. Instead, the signal must be linearized. 1 can be rewritten into:

$$distance = a\sqrt[4]{L_n} + b \tag{4}$$

Where a and b can be fitted against a measured signal-to-distance graph. To do this, a small test setup was built with a sliding rectangular obstacle connected to one sensor. The sensor signal was measured as the obstacle was slid to specific distances along the slider. This data was then recorded and used to fit 4. However, in doing so, a problem arose, as the final fit of the fitted equation was so poor that it was unusable in practice. Instead, 5 was found to better approximate the signal response found with the test setup.

$$d_{signal} = 6.0(L_n-1)^{14} - 0.1(L_n-1)^4 - 1.5(L_n-1) \tag{5}$$

where d_{signal} is the distance of the obstacle from the sensor in centimetres.

2.2.2 Placement and Wiring

The robot must be able to see obstacles in a 360° arc around its central spine. As each sensor only has a limited Field Of View (FOV), we need multiple sensors to be able to measure this full arc. However, each sensor needs to be powered and read for it to be of use. This means that there is a balance that must be attained within the design. Too few sensors and obstacles might be missed. Too many sensors and the wiring will be impractical to fit within the robot. Initial experiments with four sensors on a single segment showed that there were large gaps in the overall sensed area, and it was therefore chosen to have six sensors placed on each of the robot segments. This means that the robot has 24 total sensors placed on it.

The robot should have few wires, as each wire increases the rigidity of the system. Furthermore, as endoscopes have limited internal space, additional wires might be hard to fit. Each sensor has four total pins: two for the LED and two for the LRD. The LED has a signal pin, whose current decides the LED's brightness, and a ground pin. The LRDs were connected in "photoconductive mode" [28], and as such, they have a positive voltage pin and a signal pin.

This brings the total number of pins to 96. However, each ground and positive pin is the same. If these are connected, we are left with 48 individual connections. A few approaches are possible to reduce this number further. In other work featuring a similar design goal[15], it was chosen to process the signals of each sensor on the disk and then transmit the result using I2C. While this is an elegant solution, it requires a space on the disk to place the ICs that would handle the signal processing, and this space is not available due to the size limitations imposed by our design requirements.

Instead, it was chosen to have sets of sensors share signal wires. To be able to do this, we can use the way the sensor works to our advantage. Each sensor only works if both the LED is on and the LRD is measuring a signal. If the signal of the LRD is not captured, it does not matter if the LED is on or off. And because of 2, if the LED is never turned on, the signal will be 0. This means that multiple sensors can share the same signal lines as long as each sensor has a unique combination of signal lines, so a minimum of 10 signal wires in total is necessary for all the sensors to function. This, however, would not be ideal as this means that 2 LRDs or two LEDs on a single segment would be turned on at one time. Instead, it was chosen to have 12 signal wires, so that each no sensors on a segment would share signal wires.

By reducing the wires in this manner, signal crosstalk might occur if two sensors are close together and the LED of one is on whilst the LRD of the other is also on. To minimize this, the robot was constructed in such a way that the LED and LRD of each signal wire are as far away from one another as possible unless they combine into a sensor.

The sensors can then be read by the rest of the robot by turning on a specific LED signal wire, fol-

Table 1: The table shows the LRD wires that correspond to each LED wire. Selecting a number combination present in this table enables a specific sensor.

LED	**0**	**1**	**2**
LRD	3,2,5,4	5,0,3,1	4,1,0,3

LED	**3**	**4**	**5**
LRD	1,3,2,5	2,4,1,0	0,5,4,2

lowed by reading the 4 LRD signals that are relevant to that wire, as shown in table 1. The robot will continuously measure its surroundings whenever it is turned on, as fast as the system is able. Because of some time delays that are required for the proper functioning of the code, this results in a sensor rate of $\sim 110\ Hz$.

As this sensor rate can be higher than the calculation speed of the controller, it is possible for the Sensors to the surroundings of the robot multiple times before the results are necessary. In this case, the sensor results are averaged to minimize some of the noise seen in the sensor signal response. If the sensor is instead slower than the controller, the controller will use the last previously sent sensor signal for its control input, to ensure that the robot control does not need to wait for the sensors.

2.2.3 Disks and Spacers

Knowing the design of the sensors, we can construct disks and spacers. These disks are both the housing for the sensors, as well as the connection point of the tendons, used for the actuation. The spacers help guide the tendons along the spine. Both the disks and spacers were manufactured using a laser cutter as their 2D geometry was well suited for this fabrication method. Acrylic was chosen as it is lightweight, durable, and easily cut with the laser. During prototyping, plywood was also noted as a viable alternative to acrylic. However, it was more likely to contain certain defects, notably holes in the material, which could reduce the strength of the disk or spacer. The disks and spacers were connected to the central spine using hot glue.

The disk in figure 1.C.1 contains a central hole 0.5 mm larger than the diameter of the spring. This was done to both ensure easy assembly, and to leave some space so that the glue could get a good bond with both the disk and the spring. 4 cutouts were spaced evenly across the disk, which allows the routing of wires through the disk. On the outside rim, 12 holes are placed evenly around the disk, one for each tendon of the robot. Routing each tendon into a separate hole prevents tangling. As the robot has no internal knowledge of the actual tension on each of the tendons, it cannot know if they are slack or not. In an ideal scenario, this would never occur. However, if the robot is touching the environment, or if its bending is not ideal, the equations in 6 do not hold, and the tendons become slack. It also prevents friction between wires, as they are not in contact with each other.

On the outer rim of the disk, 6 sets of semicircular cutouts are present that connect the disk to the sensor assembly. Two semicircles were chosen instead of one to ensure a larger surface area between the two components as they are press-fitted together whilst simultaneously ensuring that enough material is present around the holes through which the tendons pass.

The spacers are identical to the disks, lacking only the 6 sets of semicircles as no sensors are connected to the spacers.

The sensor assembly that is placed on the disks comprises of four structural components and 12 sensor components(6 LEDS and 6 photo-diodes).

Figure 1.C.2 shows the first two components, made of 3d printed PLA. They are mirrors of each other and can be press-fitted onto the disk, guided by semicircular extrusions on the inside of the component. The overhang of these components around the disk keeps the signal wires within the robot, protecting them.

The first two components contain holes in which the 12 sensors are placed, as seen in figure 1.C.3. These holes protect and electrically isolate the sensors' cathodes and anodes. This design allows the sensors to be easily replaced if they break, as they can be lifted out of these holes and de-soldered from the wires. The groove on the outside allows for an easier method to connect the photodiodes in parallel by laying the positive wire in that groove. Only one groove is needed, but to simplify manufacturing, both mirrored components contain the groove.

In Figure 1.C.4 the last two components of the assembly can be seen. These were also made of 3D-printed PLA. Their purpose is to clamp the

sensors so that they point in the same direction and cannot move. They also protect the wiring and further electrically isolate the cathodes and anodes of the sensors. It also gives the overall design a smoother exterior. They are press-fitted onto the inner disk assembly, and during testing, this method was deemed to be satisfactory, as they do not fall off during trials, even when interacting with the environment.

During testing, it was observed that the overhang of the end-effector of the robot would sometimes get stuck to the edges of the obstacles after trials and that the wiring could come in contact with these obstacles, risking damage. For this reason, one of the outer shell components was expanded to have a hemispherical shape to give the end effector of the robot a more conical appearance. This successfully protected the wires and prevented the robot from getting stuck

2.3 Controller

To create a control scheme that would function well within the given situation, we need to take a look at what information we can gather from the robot and its environment, as well as what the desired task entails. The task is to move through a tube-shaped environment without hitting or damaging the walls during an endoscopy. This gives us the following constraints:

- The control must be in real-time, which means that the data collected from the robot must be processed in a short time frame [22] of 25 Hz [23].
- The control needs to be able to individually control the separate segments simultaneously.
- The only inputs are the sensor output values and the previous control positions.

Within the literature, control within human-robot interaction is generally based on allowing a rigid robot to perform a task whilst avoiding certain areas within its environment [14, 29]. This way of thinking is not applicable in this work, for a few reasons. Firstly, the environment of the robot is very constrained, with obstacles surrounding it (nearly) entirely. Secondly, the robot itself does not need to move its end effector to a certain precise position in three-dimensional space, its main task is to avoid obstacles as the entire robot arm is moved forward or backward.

For these reasons, it is better to redefine the problem. The desired control task of the robot is that it needs to be able to conform to a changing constraining environment, in real-time. For this, two control approaches were identified. The first method is similar to the rigid robot control method described in [29]. It is a greedy control algorithm whereby each segment only considers the sensors present on that segment to calculate a position that avoids its environment. As the method to calculate this position uses a PID controller, we call this method the PID controller. The second method instead considers the responses of all sensors to calculate the positions that avoid the environment for all the segments simultaneously. As the method to calculate these positions uses a kinematic model, this method is called the Model-Based controller (MB).

Both methods are implemented onto the same robot to find what control method would work best, both in simulation and on the physical robot.

2.3.1 PID Controller

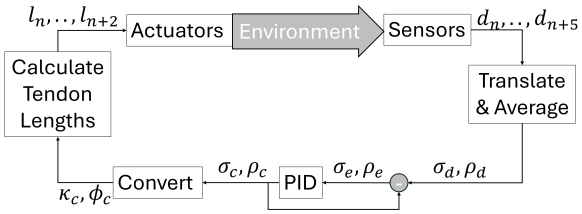

Figure 2: The full control loop of the PID controller.

As each segment is controlled individually, we only need to consider the position of a single segment. We can describe the current position of that segment with ϕ_c and κ_c, where the former describes the angle by which the segment deflects from 0 to 2π, and the latter describes the amount that the segment deflects away from the centre line of the segment. If we have three tendons which control the position of the segment placed equidistant around the centre axis, we can write the following set of equations of the length of each of these segments to constrain the movement of this

specific segment [5]:

$$l_1 = \kappa_c sin(\phi_c + \frac{-0.5 + N_{seg}/2}{3} 2\pi) \quad (6)$$

$$l_2 = \kappa_c sin(\phi_c + \frac{1.5 + N_{seg}/2}{3} 2\pi) \quad (7)$$

$$l_3 = \kappa_c sin(\phi_c + \frac{3.5 + N_{seg}/2}{3} 2\pi) \quad (8)$$

where $N_{seg} = 1, 2, 3, 4$ indicates the segment. These equations ensure that the sum of the length of the tendons stays constant, regardless of the chosen ϕ and κ. Therefore, the robot workspace can be defined as a polar coordinate system, where ϕ and κ are the angle and magnitude.

Performing PID control with polar coordinates will cause undesired control movements for our robot design. The coordinate system is therefore converted from polar to Cartesian coordinates σ and ρ:

$$\sigma = \kappa \cos(\phi) \quad (9)$$

$$\rho = \kappa \sin(\phi) \quad (10)$$

The controller input can be defined as the average of the distances d_i measured by each of the sensors per segment $i = 1, .., 8$. We use A rotation matrix to account for the two-dimensional rotation of each sensor as follows:

$$\begin{bmatrix} d_{\sigma i} \\ d_{\rho i} \end{bmatrix} = R_i \begin{bmatrix} d_i \\ 0 \end{bmatrix} \quad (11)$$

with which we can calculate the average q_d, which is our desired control position defined in σ and ρ. We can then subtract the current robot configuration q_c. This gives us an error e. This is then passed through a PID controller, which was tuned so that the system would not move too fast or erratically. The output will be the new q_c. this value will need to be converted back into ϕ and κ by converting it back into polar coordinates:

$$\kappa_c = \sqrt{\sigma_c^2 + \rho_c^2} \quad (12)$$

$$\phi_c = atan_2(\sigma_c, \rho_c) \quad (13)$$

These values can be input into 6 to move the robot to the new position. The servos are run using their own internal PID control that moves them to the new position. It is not possible to retrieve the position of these servos at any time instance which makes direct control of the servo impossible. Instead, it is assumed that by continuously updating the position of the servo with positions where Δq_c is small, the servo has time to move to the approximate new location that is desired before a new q_c is given. Furthermore, these servos can dynamically change their desired endpoint, which means that if a new desired position is calculated, the robot will instantly try and move to this new point.
The full PID control scheme can be seen in figure 2.

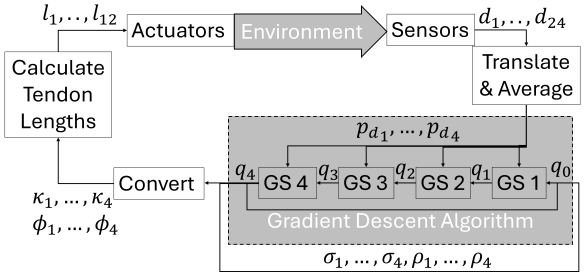

Figure 3: The full control loop of the model-based controller. GD1 to GD4 are all the gradient descent sub-steps of the controller, one for each segment

2.3.2 Model Based Controller

We can expand the equations found in 6 to create the Constant Curvature model as described in [5]. The model describes a transformation matrix:

$$T(\kappa, \phi, s) = \begin{bmatrix} \cos\phi\cos\kappa s & -\sin\phi & \cos\phi\sin\kappa s & \frac{\cos\phi(1-\cos\kappa s)}{\kappa} \\ \sin\phi\cos\kappa s & \cos\phi & \sin\phi\sin\kappa s & \frac{\sin\phi(1-\cos\kappa s)}{\kappa} \\ -\sin\kappa s & 0 & \cos\kappa s & \frac{\sin\kappa s}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

where s is the length along the curve. This curvature is capable of approximating the kinematics of one segment. Each following segment can be calculated as a kinematic chain of the previous segments where $n = 1, 2, 3, 4$.

$$T_n = T_1 ... T_{n-1} \quad (15)$$

Using these matrices we can entirely describe the approximate shape of the robot. Furthermore, we can use these matrices to describe the sensor responses in this three-dimensional space as follows:

$$\begin{bmatrix} d_{xni} \\ d_{yni} \\ d_{zni} \\ 1 \end{bmatrix} = T_n R_i \begin{bmatrix} 0 \\ d_i \\ 0 \\ 1 \end{bmatrix} \quad (16)$$

where the sensors is described as $i = 1n, ..., 6n$ and R_i as three dimensional rotation matrix in the z axis. With this equation, all sensor responses can be defined

in three-dimensional space. The overall desired position can then be calculated by averaging the sensor responses per segment, resulting in a three-dimensional q_{dn}.

In any case other than $d_{1n} = d_{2n} = ... = d_{6n}$, the new desired position will lie outside of the workspace of that segment, and while multiple segments can be used to allow a single segment achieve some of these positions, a solution is not possible in the case where all four segments must reach an unreachable position. Therefore, the controller must be robust against unreachable desired configurations.
This was achieved using gradient descent. For this, a minimizable loss function is required. We can define a general loss function for a multi-segment robot with n segments as:

$$L(q_{dn}, q_{cn}) = |q_d - q_c| \tag{17}$$

$$q_d = \begin{bmatrix} q_{d1} \\ q_{d2} \\ ... \\ q_{dn} \end{bmatrix}, q_c = \begin{bmatrix} q_{c1} \\ q_{c2} \\ ... \\ q_{dn} \end{bmatrix} \tag{18}$$

We can calculate the full set of desired positions q_d as previously discussed for any robot shape, as long as that shape is known. q_c is the current position of the final disk of each segment, described in x, y, z. We can calculate this position using:

$$\begin{bmatrix} q_{c1} \\ q_{c2} \\ ... \\ q_{dn} \end{bmatrix} = \begin{bmatrix} T_1 & 0_4 & 0_4 \\ 0_4 & \ddots & 0_4 \\ 0_4 & 0_4 & T_1 \dots T_n \end{bmatrix} \begin{bmatrix} j \\ j \\ ... \\ j \end{bmatrix} \tag{19}$$

with $j = [0, 0, 0, 1]^T$. This allows us to fully define the loss function with known variables as we know the robots' positions in κ_n and ϕ_n, and q_{dn} can be calculated using 16. To enable us to then minimize this loss function, we can use gradient descent.

This once again brings the issue of the polar coordinate definition of κ_n and ϕ_n. If we take $n = 1$ and plot the loss function, as seen in figure 4 A, we can see that the loss function contains multiple local minima. Though these minima are identical in value, they will result in finding solutions such as $(\kappa = a, phi = b + 2\pi k) \vee (\kappa = -a, \phi = b + 3\pi k)$. By using 9 as substitutes for κ_n and ϕ_n in function 14, the loss function is redefined to be dependent on Cartesian values for robot position. In figure 4.B, it can be seen that this redefinition shows a convex loss function for a single segment case, which converges to a single $\kappa = a, \phi = b$ solution. Furthermore, it ensures that the robot moves in "straight" lines through its workspace, which is important if the steps of the gradient descent are used as a control output. The final loss function for four segments

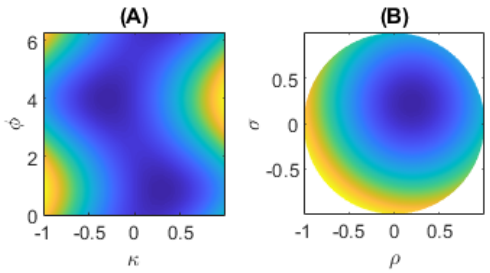

Figure 4: A shows the results of $L(\kappa, \phi, p_1)$ B shows $L(\sigma, \rho, p_1)$, where p_1 is an point is 3D space. We see that A shows a wavelike shape, a result of the sinusoids in the loss function. B instead shows a convex shape, which is preferable.

can then be defined as:

$$L(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \rho_1, \rho_2, \rho_3, \rho_4, s) = |P_d - \begin{bmatrix} T_1(\sigma_1, \rho_1, s) & 0_4 & 0_4 \\ 0_4 & \ddots & 0_4 \\ 0_4 & 0_4 & T_1(\sigma_1, \rho_1, s) \dots T_4(\sigma_4, \rho_4, s) \end{bmatrix} \begin{bmatrix} j \\ j \\ ... \\ j \end{bmatrix}| \tag{20}$$

with s being the length of a segment. The gradient can then be calculated as follows:

$$\nabla L(\sigma_1, .., \sigma_4, \rho_1, ..., \rho_4) = \left[\frac{\partial L}{\partial \sigma_1}, \dots, \frac{\partial L}{\partial \sigma_4}, \frac{\partial L}{\partial \rho_1}, \dots, \frac{\partial L}{\partial \rho_4}\right] \tag{21}$$

This gradient is used to minimize the loss function using momentum gradient descent [30]:

$$q_1 = q_0 - \eta \nabla L(q_0) - p \tag{22}$$

$$p = \eta \nabla L(q_0) \beta \tag{23}$$

with η as step-size of the gradient descent, and β as the momentum constant. This system would converge to a solution, However, when visualizing specific 2D slices of the 8D loss function, as seen in figure 5, it can be seen that the system is not a single concave solution as seen in 4.B. This might cause local minima to be found instead of the global minima that is desired. In figure 5, it is seen that the loss function is more chaotic for variables of different segments(A and B) than for variables of the same segment (C and D). It was therefore chosen to separate each segment's gradient function, while keeping the global loss function. Each segment gradient descent has a few steps that it performs, after which its resultant q_1 is used as the starting point for the next step. This also ensures that the gradient descent more closely follows the movement principles of the real robot configuration. If we run the 8D gradient descent and the

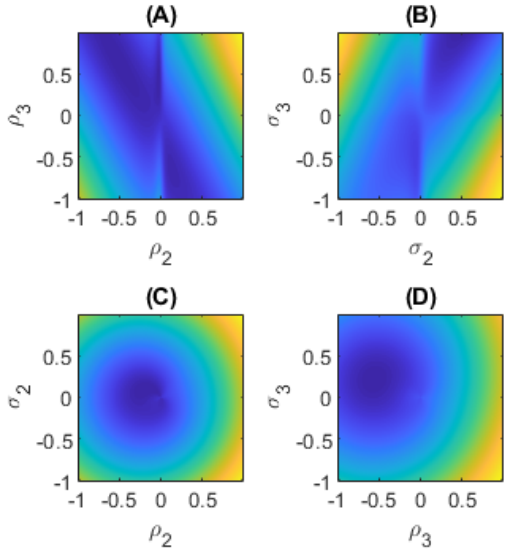

Figure 5: Four plots showing the same loss function for different sets of variables, with all others constant. in A, the ρ's of segments two and three are plotted. In B, the same is done for σ. C shows ρ and σ of the second segment. D shows the same, but for the third segment. We see that A and B are more chaotic than C and D.

segmented gradient descent for the same set of 10,000 tasks with the same allowed computation time for each task, we see a statistically significant decrease in loss of 5.4% ($p = 9.6 \cdot 10^{-7}$) for the segmented method, which shows that this method is better at finding a minimal loss configuration for the robot. The full control method that is used can be seen in figure 3.

2.4 Electrical Design

For the robot to function, the electrical design of the robot must include the following subsystems:

- A processor is required that can perform complex mathematical control operations at 25Hz [23].
- Electronics that can transmit digital and receive analog signals from the sensor array on the robot arm.
- Electronics that can generate and transmit PWM signals to the servo motors.

These subsystems must be connected using low-latency methods such as GPIO, I2C, or SPI so that real-time control of the robot is possible. To simplify the design, the processor that runs the controller should also handle the sensor array and motor subsystems. Based on these requirements the design shown in figure 1.A was developed. In grey, the signal path from control output to motor movement is shown. In green and purple, the communication with the sensor array is shown.

2.4.1 Central Processor

The central processor can thus be stated to have two requirements. It must contain low latency IO functionality, to be able to control the subsystems, and it must be able to have a processor powerful enough to perform the required control calculations and the switching at the same time. Initial experiments were performed using the Arduino Mega. which meets these requirements. However, the Arduino board only has 248kB of storage [31], which is lower than the required 1.86 MB needed to store the compiled robot control. Another problem is that the Arduino Mega is single-threaded. To have the control and sensing run simultaneously multi-threading is preferable.

Instead, it was chosen to use a Raspberry Pi 4b (RPI). The RPI has 16 GB of static storage, which is more than enough to run the robot control program. The RPI is capable of running a full Linux operating system. This allows the usage of C++, with many common libraries, simplifying the development process. Using the GUI capabilities of the RPI, the process of development was simplified, as a debug graphical interface was made that was able to show relevant sensor, control and actuator values. This allowed for "software in the loop" testing of the control, which vastly increased the speed of implementation of the robotic control onto the RPI. A downside of the RPI over the Arduino Mega, is its limited IO capabilities. It lacks an ADC and only has 2 PWM signals. Therefore, external Integrated Circuits (IC) were required.

2.4.2 Sensor Circuitry

To utilize the sensors, two things are needed: A transmitting signal to power LEDs and receiving a signal to measure the LRDs. For the LEDs, it is theoretically possible to directly control them using the General Purpose Input Output (GPIO) pins on the RPI, however, they only support limited voltage and current, which would severely limit LED brightness. Using a separate analog signal instead allows calibration of the LED brightness to optimize measurement accuracy. To switch this analog signal, an Analog Multiplexer(AM) is used. The LEDs are connected to this multiplexer in series, as seen in figure 6. This ensures that all LEDs receive the same current, as the LEDs might otherwise get damaged. The AM also has one of its pins connected to a single red LED. This LED is led is turned on when the sensors are turned off, to ensure that the system is functional. The AM has a resistance of 150 Ω [32], which limits the brightness of the LEDs at 5 volts analog input. This resistance was reduced to $50\ \Omega$ by connecting 3 AM in parrallel. The AM is controlled with 3 IO signals, connected to the RPI.

The LRDs need to be read using an ADC. During testing with Arduino, it was observed that 10 bits is enough precision to be able to measure the signal response of the LRD. So a 10-bit ADC was chosen, which

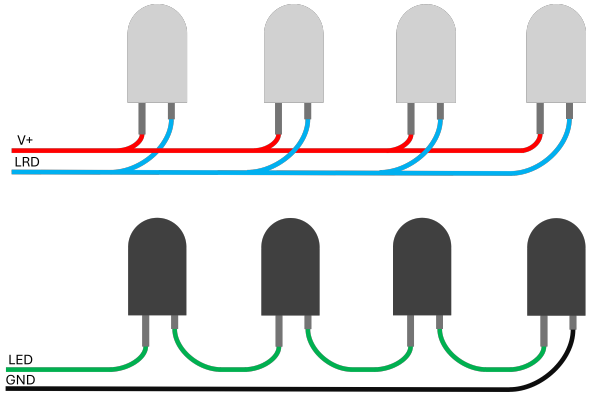

Figure 6: Figure showing the photodiodes in white, the IR LEDs in black, and the way they are wired. The robot contains 6 sets of these wires.

allowed measuring all 6 data channels [33]. The LRDS are connected in parallel, in photoconductive orientation. They are connected to the ADC with a 1 $M\Omega$ resistance voltage divider. The ADC measures the voltage of the LRD with the highest conductivity, which is proportional to the amount of light shining on that LRD[28]. The ADC is connected to the RPI using the Serial Peripheral Interface (SPI).

Both these systems work better at 5V, compared to the 3.3V of the RPI. For the ADC a higher voltage means a larger range that can be measured by the ADC, as well as an improved communication speed[33]. For the LEDs, this means a higher maximum analog voltage [32]. To allow communication between this subsystem and the RPI, voltage converters were used that allowed for uninterrupted SPI communication.

2.4.3 Motor Circuitry

The position of the motors is controlled with a PWM signal. These signals can be generated using an external driver board that is connected to the RPI using the Inter Integrated Circuit (I2C) communication. This allows the RPI to send signals as desired to the PWM driver, who then handles transmitting this PWM signal until a new signal is desired. Both the driver and motors accept 3.3V as a signal strength, and as the range of motion is dependent on pulse width, and not voltage, no accuracy is lost. As such no voltage conversion is required. The motors themselves are powered using a separate 6V power line. This is the maximum voltage allowed for the chosen servos.

2.5 Experimental Design

To experimentally evaluate and validate the robot's performance, each subsystem must first be independently assessed. The subsystems that require this assessment are the controllers and sensors.

2.5.1 Simulated Controller Performance

The performance of the controllers was estimated using simulated control tasks. For this, a set of random κ_d and ϕ_d were generated. We use the CC model 14 to calculate a robot configuration from this, which gives us a set of q_d. Both controllers were then tasked to get as close to these reachable points as possible. The simulation was run with 5000 sets of random κ_d and ϕ_d values to ensure the reliability of any measured differences.

2.5.2 Experimental Setup

Both the sensor and the full robot experiments used a Universal Robot 5e™(UR) to facilitate data acquisition. The end effector of this robot was controlled using a MATLAB script that was able to send a series of positions to the UR. It would then move its end effector to the designated position. After the movement was completed, the same MATLAB script would get the sensor response and position data from the continuum robot. During the experiments, it was ensured that the robot position was static before information was recorded to ensure that the position data of both robots was accurate.

2.5.3 Sensor Experiments

For the sensor experiments, a white orb with a diameter of 10 mm was used as the sensor target. It was connected to the rest of the robot with a matte black screw with a length of 40 mm, to ensure that the orb was the only thing that the sensors would measure. Two different movements were used to map the sensor response, to be able to estimate the sensor range and the sensor FOV. The starting configuration of both the UR and the robot can be seen in figure 7.A.
The first experiment moved the UR in a semi-cylindrical motion starting at the tip of the continuum robot. The UR moved the target in a 180-degree arc around the robot, close to the sensors, and after that arc was completed, it would move up slightly and repeat the arc in reverse. This happened until the UR reached the base of the continuum robot. In the second experiment, the UR moved in a semicircular motion, whose radius increased. This motion was started at the centre line of a disk.

2.5.4 Robot System Experiment

Quantitative and qualitative experiments were devised to test the functioning of the entire robot system. It was not viable to develop a test setup that could quantitatively estimate the performance of the robot in a fully constrained environment, as the ground truth of the robot orientation would not be able to be calculated or measured. Inertial Measurement Units (IMU) are too

impractical, as they would not easily fit inside the device. Furthermore, they require a lot of post-processing and calibration to estimate an accurate position, which would fall outside the scope of this research. Visually measuring the positions using motion capture is similarly unattainable. Motion capture systems use IR retro-reflections to estimate the position of a target, which would be blocked by the constrained environment. The motion tracking would also be influenced by the proximity sensors on the robot itself, causing incorrect measurements.

Instead, it was chosen to quantify the effectiveness of the control methods by looking at how the robot responds to a single obstacle. The position of this obstacle can then be accurately controlled with the UR. This obstacle, a white plane, was moved towards the robot in small steps. After each step, it was evaluated if the robot was in contact with the plane or not. If contact was observed, the travelled distance was recorded, and the experiment was reset. In figure 7.B the test setup can be seen, where the white plane obstacle is positioned next to the fourth segment of the robot.
This was done in three ways. One where only the sensors on the fourth segment would be able to detect the plane, one where only the third segment would detect the plane, and one where both the fourth and the third segment could detect the plane. These three tasks highlight the different capabilities and constraints of the robot.

For qualitative experiments, a smooth white pipe segment with a diameter of 7.5 cm was additively manufactured, as seen in figure 14.B. This tube contained a small door, which made it possible to see the position of the robot within the tube at the end of the experiment. This was then placed onto the end effector of the UR, which enabled the positioning of the tube in three-dimensional space. The curve of the tube was chosen so that the robot would fit inside in its straight configuration during initialization. After the controller was turned on, the tube was rotated to constrain the robot.

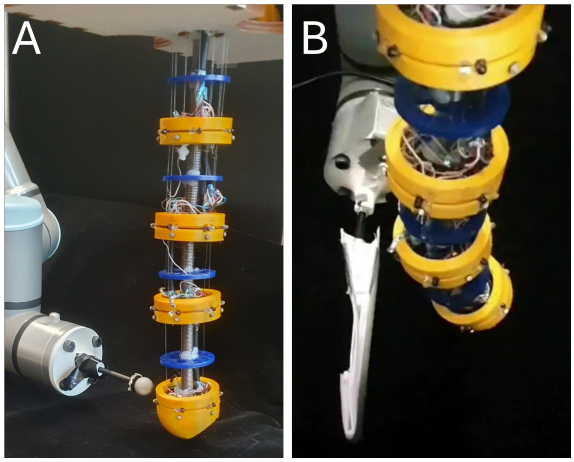

Figure 7: Images showing the experimental setups. A shows the sensor experiments, where a white point target was moved along the static robot. B shows the obstacle avoidance experiment, where the obstacle is the white plane. This photo was taken while the robot was turned on, and as such, it has curved away from the plane.

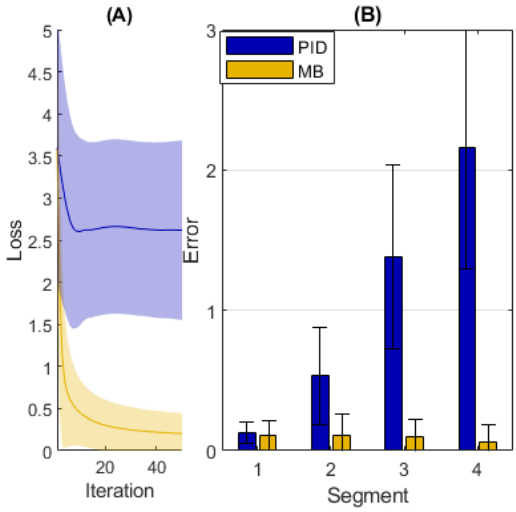

Figure 8: In the figure, the results of 5000 simulations of the control task are shown. In A, the loss function values are shown as each control method calculates for 50 iterations. In B, the error between the current segment position and the desired control position is shown at the end of each trial.

3 Results

The results from the three experiments that were conducted were separated into categories: control simulations, sensor experiments and robot experiments.

3.1 Control Simulations

In figure 8, the results of the 5000 controller simulations are shown. Both controllers ran for 50 iterations. In 8.A, it is seen that for this task, the loss of the model-based controller is significantly lower than the loss of the PID controller. The difference is almost immediate, and before 10 iterations are reached, both controllers lie outside their respective standard deviations. The average MB controller loss was reduced by 94.6%, whilst the average PID controller loss was reduced by 26% compared to the initial loss. The MB controller has an average final loss of $92.6\%(p = 0.000)$ lower than the PID controller.

In figure 8.B, the average and Standard deviation of the Euler distance between the desired control position and the simulated robot position at $iteration = 50$ is

shown. We see that the MB error stays consistent per segment, while the PID error gets progressively larger, the further along the robot the segment is. In figure 9, the results of desired and robot configurations at $iteration = 50$ of a single trial are shown. It shows that the first two segments of the PID robot output are similar in shape compared to the desired configuration. However, the last two segments are far away from the desired positions. The MB controller more closely follows the desired configuration for all segments. During

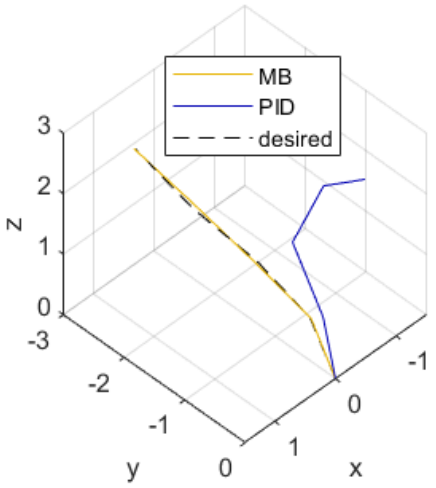

Figure 9: The figure shows the controller output after 50 iterations for one of the 500 control trials. The dashed line is the desired output, the yellow line is the MB controller output, and the blue line is the PID output.

another test of 5000 trials, this time for 100 iterations, the computation time of each trial was recorded. These results can be seen in table 2. The measurements were performed using MATLAB 2024, running on an Intel i7 9th gen CPU, with 16 GB of RAM. Note that this is not the same hardware as used for the final robot. We see that the PID requires a shorter calculation time than the MB controller.

Table 2: temporal results of 5000 trials. Each trial is 100 iterations long. Run on MATLAB2024, Intel i7 9th gen CPU.

	PID	Gradient
Total computation time [s]	1.1925	993.9272
Time per trial[s]	$2.3850 \cdot 10^{-4} \pm 3.5470 \cdot 10^{-4}$	0.1988 ± 0.0264
Time per iteration [s]	$2.3850 \cdot 10^{-6} \pm 3.5470 \cdot 10^{-6}$	0.002 ± 0.00026

3.2 Sensor Experiments

To achieve the results in figure 10, a white reflective target was moved around the robot in a semicircle of increasing radius. Sensors 19, 20 and 21 were the only sensors in the path of the target. This means that the results are based on these three sensors. The sensor signal of each sensor was recorded as the target passed through its centre line. These results were then averaged to get the "mean of sensor" result, with the respective SE.

Signals were also recorded when the target moved through another sensor's centre line so that a baseline sensor response could be established, as seen in figure 10. In the baseline, it can be seen that the distance of the target does not affect the sensor signal if the sensor cannot see the target. When a sensor can see the target, the sensor response of that sensor is dependent on the distance of the target to the sensor. We see that the SE of the baseline is lower compared to the sensor SE and that the sensor SE increases as the target gets closer to the sensor. The solid line seen in the figure represents the polynomial 5, where L_n is the "mean of sensor" normalized l_{signal} values. The dashed line represents the equation 4, where a and b were fitted using the "mean of sensor" normalized l_{signal} values. Figure 11 is the

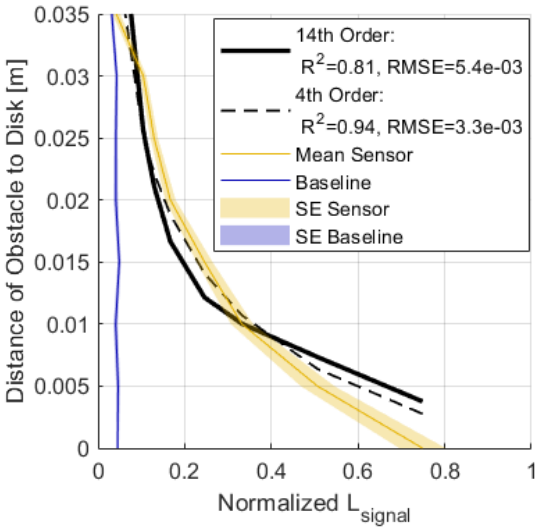

Figure 10: The mean signal response of three sensors is shown, with its standard deviation, both when exposed to an obstacle and the baseline response. The solid black line is the estimated distance used by the controller, and the dashed black line is an improved estimation, based on this fitted to this dataset.

result of the UR moving in a semicircular motion at a changing vertical height along the sensor for a total of 20 cm along the robot. This creates a semi-cylinder. In figure 11.A, the maximum sensor response is shown as the z value of the plot. We see four horizontal bands, each with three peaks of intensity. This corresponds to the robot's four segments and the three sensors on each of these segments. Only three sensors are seen, as the

target only moves around half the robot, $0°$ to $180°$.

Using a threshold of 0.02, the background noise is manually filtered from the rest of the signal. By looking at the horizontal lines in figure 11.B the cumulative angle of the semi-cylinder that is visible to the sensor can be calculated. This peaks at approximately $110°$ for each segment, but quickly falls off on each side of that peak. There are large gaps in this FOV, which correspond to the areas on the robot where the spacers are located. This plot can also be used to calculate the overall percentage of this semi-cylinder that is within the view of the robot, which is 16.7%. When looking at the rows of the image, we see that 45.2% of the rows are within the FOV of the robot. Within figure 11.A, there

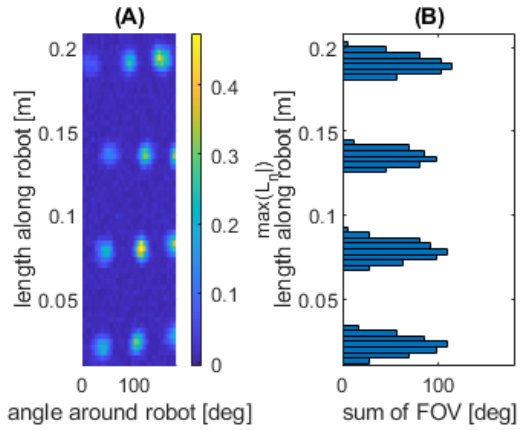

Figure 11: A shows the maximum sensor response as a target was moved around the robot in a hemispherical grid. B shows the sum of all fields of view of each sensor, separated according to height along the robot.

are eight sensor responses with their full FOV in the traversal and longitudinal axis within the dataset. This, combined with the circular motion along the horizontal axis of the robot, allows us to calculate the FOV of these eight sensors that are fully in view. The average FOV of these sensors, with their SE is $40.8° \pm 1.2°$.

3.3 Robot Experiments

In figure 12, the results of the obstacle avoidance experiment are shown. Each of the tests was repeated 5 times. The static test is the control, whereby no controller is turned on, and the robot is placed in its initial position. Therefore, the deviation seen in this bar is fully caused by observation error, as the test relies on human eyes to detect the hit. We see that the MB controller performs better than the PID controller for the fourth segment test, increasing the travelled distance by 92.9% ($p = 1.0 \cdot 10^{-10}$). The third and fourth segment test shows an increase in the robot performance of 65.5% ($p = 1.8 \cdot 10^{-5}$). The second experiment instead shows a decrease of performance of 23.5% ($p = 1.9 \cdot 10^{-6}$). P values were calculated using 2 sample T-tests.

Furthermore, we see that the standard deviations are low for all tests, aside from the MB controller test for both segments. Visual observation of the robot during the first and third tests showed that hits on the robot occurred on the inner rim of the disk rather than on the sensor itself. For the second experiment, the obstacle hit the sensor itself.

In figure 13, the mean MB controller response is

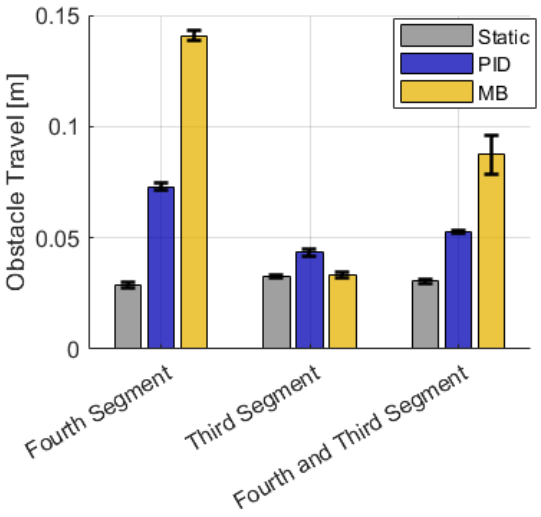

Figure 12: A figure showing the distance a 10 cm long and 5 cm wide obstacle could travel along the Y axis towards the robot before the robot was hit. For the fourth segment experiment only the sensors on the fourth segment could sense the obstacle. For the third segment experiment, only the third segment could sense the obstacle. For the Fourth and Third Segment experiment, both segments can sense the obstacle.

shown. We see that in all cases multiple segments are all curved, indicating that the MB controller uses multiple segments to adapt to the environment. Each of these curves is in three dimensions. However, as the curves were for the most major part in the $y+$ direction, only this was shown.

Figure 14.A shows that the robot design itself is capable of orienting itself into complex three-dimensional shapes. Figure 14.B shows the result of the qualitative robot experiment, whereby the robot was able to configure itself inside the tube. The controller that was turned on during this experiment was the MB controller.

4 Discussion

As the multiple components of the robot were individually constructed and experimentally assessed, it makes sense to individually examine where these components

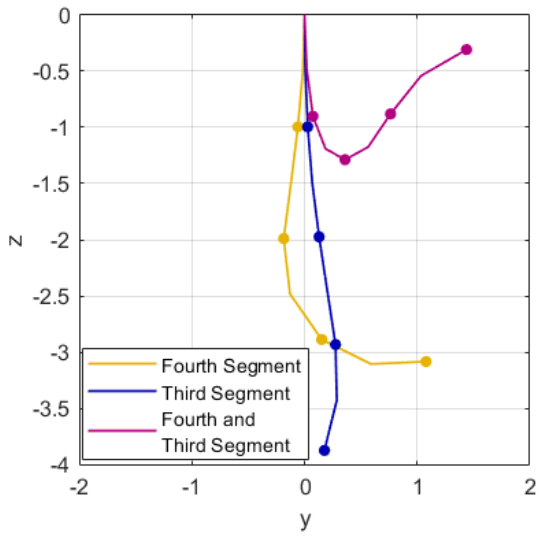

Figure 13: A figure showing the mean of the MB controller response for each of the tests. The obstacle was placed at the Y- side of the robot, moving towards Y+.

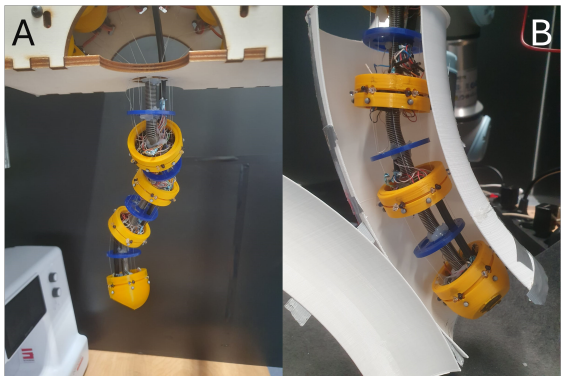

Figure 14: A shows an image showing the robot positioned in a complex configuration. This was done with arbitrary κ and ϕ values that were sent to the servos and not due to a control input. B shows an image of the robot partially constrained in a curved tube with a diameter of 7.5 cm. The pose was calculated by the robot controller based on the sensor signal. When a steady-state was reached by the controller, it was turned off, and the side panel was removed to take the image.

succeed and where improvements in the approach can be made. At the end of the discussion, it is assessed if the requirements set out in the methods are achieved.

4.1 Sensor

The results of the optical sensors in figure 10 show that there is a clear separation between baseline sensor signals and excited sensor signals up to a distance of 35 mm from the sensor. This means that each sensor can measure obstacles for distances up to half the length of the largest lumen. Or, if the robot is in the centre of the largest lumen, to detect any of the walls. Furthermore, we see that the standard error is low, which indicates that all individual sensors have approximately similar sensor responses and follow a similar curve.

We see that the variance of the baseline is low, which indicates there is no hysteresis in the sensor, which was desired and expected based on the method by which the sensor functions[28]. Furthermore, it indicates that when one sensor is excited by an obstacle, another nearby sensor will not also be excited. This means that there is no sensor crosstalk, which shows that the wire reduction method works as expected.

Within figure 11.A, it can be seen that there is a variation in the signal intensity of the sensor response, even though all sensors should be at the same distance and target. This might be caused by imperfections in the sensor assembly. Each sensor is constructed individually, as seen in figure 1.C, and small differences in that construction can contribute to sensor baseline variations. In figure 10, min-max calibration was used on the measurement data before the sensor responses were averaged. The low variance in sensor response in this figure shows that the min-max calibration is a viable method to improve sensor consistency.

If these results are compared to the results of this experiment with the capacitive sensors[16] developed for endoscopy, some differences can be noted. The capacitive sensors have a range of approximately 10 mm, 3.5 times less than the optical sensors. This suggests that these sensors are less likely to be able to fully measure larger lumena. The FOV of the capacitive sensors is 78°, which is 1.9 times wider than the FOV of the optical sensors. This would mean that fewer capacitive sensors are required to fully estimate the surroundings of the robot, simplifying the robot design. A main shortcoming of the capacitive sensors compared to the optical sensors is the parasitic capacitance. This parasitic capacitance will mean that control outputs will be based on incorrect information, limiting the overall robot performance.

Comparing the optical sensors to TOF sensors [15], the opposite can be seen. These sensors have an effective range of 130 mm, with a FOV of 16.5°. The increased range of the TOF is useful in many scenarios but unnecessary within the lumen, as this range is more than double the diameter of the largest lumen. The limited FOV is also problematic, as this would require 2.4 TOF sensors for each optical sensor to achieve the same FOV. One of the main limitations identified for TOF [15] sensors was the slowdown of the signal processing when more TOF sensors are used, which indicates that using more TOF sensors to estimate the same area is undesirable.

4.1.1 Lumen Structures

Using literature, the effectiveness of the sensors in detecting structures present on the walls of the lumen can be quantified. The stomach walls contain pits with a depth of $0.2mm$ with a diameter of $70\mu m$[34], The depth of these pits falls within the SE range as seen in figure 10, which means that the sensor will not detect these pits, and it will instead measure the stomach wall as a smooth wall. For the small and large intestine, the surface has many villi, which are $0.5-1\ mm$ long hair-like protrusions on the surface. The sensor will also not be able to detect these and will measure the walls of the intestine as smooth curvatures.

Larger structures that are present in the small intestines are the folds of Kerckring. These folds are generally traversal and are either circular or crescent-shaped [35]. As there are six sensors, with each having a FOV of $40.8° \pm 1.2°$, the FOV gaps between the sensors are at most $20.4°$. So unless the crescent arc is shorter than $20.4°$, the fold will be detected by the sensor. The distance between these folds is 0.0036 to $0.0064\ m$.[35]. When comparing this to the height of the sensor view, $0.023\ m$, it can be seen that the next fold would be inside the FOV of the robot before the last fold leaves its FOV.

4.1.2 Distance Function Fit

When plotting the results of 5 against the mean measured distances, as seen in 10, we can see that the final fit is poor. If we instead look at a 4th-order approximation in the same figure, we see a closer fit to the curve of the sensors. The reason for the discrepancy between the approximation found in the test setup and the approximation found in the final robot is found in the way the test setup was constructed, the rectangular target, and the slider that was attached to it.

When close by, the large square would reflect more light towards the sensor than the smaller point obstacle used in the final experiment. This means that the signal would, in turn, increase faster than the theory would suggest as the obstacle moved closer to the sensor. When the object was further away, the slider would reflect some of the light towards the sensor. This light would cause the signal to decrease less than the theory would suggest. This fully explains the shape difference between the initial polynomial and the improved polynomial, as seen in figure 10.

The effects of this error on the controller are only minor. The signal is averaged, and the error is present in all different directions, so they cancel out. The current polynomial reacts more weakly to changes between 1 and 3 cm in distance from the robot, whilst it responds more strongly to movements within one cm from the disk. As we only tested obstacles from one side within the experiments shown in figure 12 and measured only the moment at which collision occurred, there is no method by which the badly fitted polynomial would change the values of these outcomes.

4.2 Robot

The performance robot and its control can be discussed by looking both at the simulated controller performance, as well as the physical robot performance. Both elucidate various concepts and effects, allowing us to accurately assess the successes and areas of improvement of the overall robot system.

4.2.1 Simulated Controller Performance

The experimental simulations suggest that the MB controller outperforms the PID controller in position control tasks. This is made evident by the differences in both the mean and standard deviation of the loss functions in figure 8.A. It shows that the loss reduction of the MB controller is 92.6% larger than the loss reduction of the PID controller. This follows literature, where model-based methods outperform model-free methods [36] by having lower overall tracking errors.

Figure 8.B shows that the loss of the PID increases for each consecutive segment, whilst the MB loss stays constant for each segment. This indicates that the underlying assumption made for the PID position control, where $\kappa \approx x,\ \phi \approx y$, does not hold. An example of this can be seen in figure 9, where the last two segments of the PID output deviate a large amount from the desired output when they are compared to the MB output.

In literature, shape constraints can be used to guide robot movement[37] instead of position control. These shape constraints allow for a more explicit collision avoidance constraint compared to the MB controller. There are, however, problems with this method for endoscopic applications. The human body is not rigid, and as such, the shape constraint needs to change continuously. For endoscopy specifically, measuring this shape will require external mapping such as MRI or ultrasound, which limits the applicability. The position control method used in this work circumvents this limitation by acquiring the desired robot position via the sensors placed on the robot itself.

4.2.2 Physical Robot Performance

The physical robot experiments suggest that the MB controller outperforms the PID controller in most physical obstacle avoidance tasks, as seen in figure 12. This aligns with the simulation results. The MB controller was able to generate control outputs that used the actuation of multiple segments to allow for a larger range of motion compared to the range of motion of a single segment, as seen in figure 13. Because the PID controller only actuates segments that measure the obstacle, it was limited to the range of motion of that specific segment. The reduction in performance of the ”third and fourth

segment" obstacle avoidance compared to the "fourth segment" obstacle avoidance for both controllers suggests that the range of motion of the robot itself was a limiting factor. The third segment is closer to the base than the fourth. This means that, because of the properties of a constant curvature [5], the same κ results in a smaller deflection.

Whilst the PID is outperformed by the MB controller, it still outperforms the static case. This might seem contrary to the result seen in figure 8.B and 9. However, this difference is explainable by the difference in the experiment objective. Notably, the estimated position of the PID is not relevant for obstacle avoidance experiments. Instead, the measurements of the sensors directly inform the next PID output, allowing for effective obstacle avoidance. As position-tracking itself was not required for endoscope obstacle avoidance, no such experiments were done. This complicates a direct comparison of the robot that is presented in this work with robots in the literature, such as 2 DOF [36] or 6 DOF MPC [37] robot arms, as quantitative comparisons are not possible. However, the MB controller outperforming the PID controller in physical robot experiments follows the literature, where model-based controllers outperform model-free controllers in physical robot experiments [36, 37].

The third segment experiment result in figure 12 shows a shortcoming in the robot design. The MB controller performs identically to the static robot case, with the PID controller only outperforming the MB controller by 23.5%, the lowest outperformance of any test. In particular, the MB controller result does not align with the controller output shown in figure 13, which suggests that the robot will create a shape different from a static case. This is a constraint of applying the constant curvature model as a kinematic model, which is commonly observed in the literature [37]. It can be seen that the controller output is a C-shaped curve in the ZY plane. To create this curve the third segment moves in the Y+ direction, and the fourth segment in the Y- direction. These cause antagonistic tendon actuation in the robot, which constrains the robot into a static configuration. The robot is more strongly constrained by the MB controller compared to the PID controller. This is because the PID controller does not move the fourth segment in any Y direction, which lessens the antagonistic forces. To improve the accuracy of the robot model, a redesign of the robot arm is preferable over increasing the complexity of the model. The literature [17] shows that TDCRs are capable of configuring themselves in shapes as shown in figure 13 if certain design constraints are met. On the other hand, increasing the complexity of the controller offers only minor reductions in tracking error at a large computation cost [38].

The MB controller is capable of positioning the robot in the centre of a constrained environment, as shown in figure 14. This suggests that the MB method is similarly capable of automatic endoscopic steering as automatic steering based on camera vision [10]. The MB controller is 25 times faster, with the camera-based method requiring 0.05 seconds per iteration. Other soft robotic controllers for lower DOF systems similarly show a longer iteration time of 0.03 [37] or 0.0034 [38] seconds per iteration. More complex models can take 2.8 hours to solve 60-second control tasks[38]. This suggests that the MB controller would achieve a higher control loop speed on weaker hardware, which is promising for future integration into endoscopes.

4.3 Robot Design and Construction

The image in figure 14.A shows that the robot can achieve three-dimensional curvatures. This suggests that the robot is physically capable of creating compound curves. However, aside from the tendon actuation constraints identified during the robot experiment, the robot's performance is limited by other factors of the robot's design.

4.3.1 Stiffness

The robot moves non-isometrically, which deviates from the expected isometric tendon continuum robot behaviour [36]. The movement was constrained specifically in the plane of the sensor wiring, indicating that the wiring was the cause of the movement constraint. However, as both the PID and MB robot experiments were affected by this limitation equally, the data is still valid.

4.3.2 Spring Fixation

Hot glue adhesion of the disks to the spring resulted in weakening bonding over time. These bonds could break, allowing the segment to shift along the spine of the robot. Using a set of shorter springs, mechanically connected to the disks, as shown in [15], would be more robust.

4.3.3 Robot Size

The robot's design has to be modified to fit within the 12.8 mm [19] size constraint. This has mainly to do with the sensors' size and the wiring's placement. Reducing the size of the sensors can be done by replacing the current LED and LRD with smaller versions of each, both having a width of 2 mm and height of 0.8 mm [25] or less [39]. These sensors could then be mounted on an FPC, which would wrap around the disks of the robot. The rest of the robot could then be constructed as endoscopes currently are, with the addition of the mounting location of the sensors.

4.4 Requirements Evaluation

If we look at the requirements set out in the methods and the discussed results, it can be concluded that most requirements were achieved:

1. The robot is capable of sensing human tissue as well as other opaque objects. It can measure up to a range of 3.5 cm. Each sensor has a FOV of $40.8° \pm 1.2°$. Their placement around the disk allows the robot to see 45.2% of its total observable surroundings.

2. The robot is about four times as large as a normal colonoscope, with a diameter of 43 mm. Its size will need to be scaled down 4 times for it to be an effective endoscope. Sensors[25, 39] and TDCR designs[2, 3, 17, 19] at these scales are commercially available. This means that the downscaling would require no additional scientific advancement.

3. The robot is capable of creating three-dimensional curves using its 8 degrees of freedom. However, its design is constrained by the tendon actuation and the wire stiffness.

4. The sensors can run at approximately 110 Hz, surpassing the minimum of 25 Hz. The model-based controller can run for 20 iterations and still pass the loop speed requirement, with one iteration running at 500 Hz. This means that the robot is controllable in real time.

5. The control can avoid obstacles and conform itself to a constrained, curved environment. It is only limited by the physical design constraints of the robot.

5 Conclusion and Recommendation

In this research, a novel fusion of sensing and control for endoscopic procedures was examined. It was found that the created robot is generally able to avoid obstacles and conform itself to an environment. Furthermore, model-based control methods for this robot allow this robot to be able to better avoid obstacles than compared to a PID control method, both in simulation and real-world experiments. In some situations, the controllers were constrained by the physical robot design, which resulted in inconclusive results. Overall, the results show that this method of designing and constructing a tendon continuum robot that avoids the walls of the lumen was successful. The next step in development should focus on miniaturization. Adapting a currently existing endoscope by connecting a cap to the endoscope tip that contains the sensors would be the quickest method to add sensors to currently existing endoscopes, the bendable section can then be controlled by connecting a servo to the knob on the handle of the endoscope, which would allow the controller to steer the tip without directly modifying the endoscope body. Adapting multi-bending endoscopes in this way allows more in-depth testing of the MB controller's effectiveness.

Whilst the current sensor setup is fit for purpose, further research might require an increase in the sensor field of view. To increase the traversal field of view, additional sensors could be placed on each disk. Integration within the system is trivial, as the sensors would be processed the same as any other sensor. The longitudinal field of view can be increased by replacing the spacers with full disks. The sensor responses that this would generate would only be useful for the MB controller, as it would function as a separate CC curve that shares its κ and ϕ with the segment on which the spacer is placed.

Replacing the optical proximity sensors with force sensors is also an interesting avenue of research. This would allow the robot to work in more strongly constraining environments where contact with walls cannot be avoided. The controller and sensor processing would not need to change, and hypothetically, the robot would still be able to be controlled. The main disadvantage of this modification would be that he constant curvature assumption is less accurate when the robot is in contact with the environment. This reduction in accuracy would be valuable to estimate. Comparing the proposed method of force feedback with state-of-the-art force feedback robot control systems [13] would then further the overall progress in developing autonomously actuated endoscopes.

The current system runs fully autonomously. It is, however, trivial to instead influence the position of the robot with a user control input. There are two methods, both of which have different effects:

1. Direct control: Hereby, the σ and ρ that result from the controller are changed by the user. This moves the desired segment of the robot in any direction, regardless of whether there is an obstacle in the way. It also will not try to keep the other segments in an optimal position, as the controller output is overwritten.

2. Informed control: Hereby the $[x_{des,n}, y_{des,n}, z_{des,n}]^T$ is influenced by the user. This means that the control will still optimize the robot's shape.

The suspected ideal usage of a user control method is that every segment except the end segment is controlled using informed control, with the end effect being controlled with direct control. This means that the overall robot shape will find the optimal curve for all desired points, but it gives the user the option to make precise adjustments to find the optimal angle for the region of interest, which is important for a high success rate for

endoscopy [18]. Because of the fully electronic control of the endoscope, haptic control [6] or more simple joysticks could be used to ergonomically control the endoscope.

References

[1] Peery AF, Crockett SD, Murphy CC, Jensen ET, Kim HP, Egberg MD, et al. Burden and cost of gastrointestinal, liver, and pancreatic diseases in the United States: update 2021. Gastroenterology. 2022;162(2):621-44.

[2] Geboes K, Geboes K, Jouret-Mourin A. Endoscopy and histopathology. Endoscopy. 2013;1:3-32.

[3] Schneider A, Feussner H. Chapter 5 - Diagnostic Procedures. In: Schneider A, Feussner H, editors. Biomedical Engineering in Gastrointestinal Surgery. Academic Press; 2017. p. 87-220. Available from: https://www.sciencedirect.com/science/article/pii/B9780128032305000051.

[4] Marieb EN, Hoehn K. Human anatomy & physiology. Pearson education; 2007.

[5] Webster III RJ, Jones BA. Design and kinematic modeling of constant curvature continuum robots: A review. The International Journal of Robotics Research. 2010;29(13):1661-83.

[6] Reilink R, Stramigioli S, Kappers AM, Misra S. Evaluation of flexible endoscope steering using haptic guidance. The International Journal of Medical Robotics and Computer Assisted Surgery. 2011;7(2):178-86.

[7] Waddingham W, Kamran U, Kumar B, Trudgill NJ, Tsiamoulos ZP, Banks M. Complications of colonoscopy: common and rare—recognition, assessment and management. BMJ Open Gastroenterology. 2023;10(1):e001193.

[8] Kavic SM, Basson MD. Complications of endoscopy. The American journal of surgery. 2001;181(4):319-32.

[9] Waddingham W, Kamran U, Kumar B, Trudgill NJ, Tsiamoulos ZP, Banks M. Complications of diagnostic upper Gastrointestinal endoscopy: common and rare–recognition, assessment and management. BMJ Open Gastroenterology. 2022;9(1):e000688.

[10] Deng Z, Jiang P, Guo Y, Zhang S, Hu Y, Zheng X, et al. Safety-aware robotic steering of a flexible endoscope for nasotracheal intubation. Biomedical Signal Processing and Control. 2023;82:104504.

[11] Nazari AA, Zareinia K, Janabi-Sharifi F. Visual servoing of continuum robots: Methods, challenges, and prospects. The International Journal of Medical Robotics and Computer Assisted Surgery. 2022;18(3):e2384.

[12] Vrooijink GJ, Denasi A, Grandjean JG, Misra S. Model predictive control of a robotically actuated delivery sheath for beating heart compensation. The International Journal of Robotics Research. 2017;36(2):193-209.

[13] Yip MC, Camarillo DB. Model-less hybrid position/force control: a minimalist approach for continuum manipulators in unknown, constrained environments. IEEE Robotics and Automation Letters. 2016;1(2):844-51.

[14] Ding Y, Thomas U. Collision avoidance with proximity servoing for redundant serial robot manipulators. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2020. p. 10249-55.

[15] Abah C, Orekhov AL, Johnston GL, Simaan N. A multi-modal sensor array for human–robot interaction and confined spaces exploration using continuum robots. IEEE sensors journal. 2021;22(4):3585-94.

[16] Giacoppo GA, Mayer J, Hartmann J, Bachmann AL, Pott PP. Actively shielded capacitive proximity sensor for endoscopy. In: Current Directions in Biomedical Engineering. vol. 9. De Gruyter; 2023. p. 93-6.

[17] Hamada K, Horikawa Y, Koyanagi R, Shiwa Y, Techigawara K, Nishida S, et al. Usefulness of a multibending endoscope in gastric endoscopic submucosal dissection. VideoGIE. 2019;4(12):577-83.

[18] Koo HC, Moon JH, Choi HJ, Ko BM, Hong SJ, Cheon YK, et al. The utility of a multibending endoscope for selective cannulation during ERCP in patients with a Billroth II gastrectomy (with video). Gastrointestinal endoscopy. 2009;69(4):931-4.

[19] Kay M, Wyllie R. 62 - Colonoscopy, Polypectomy, and Related Techniques. In: Wyllie R, Hyams JS, editors. Pediatric Gastrointestinal and Liver Disease (Fourth Edition). W.B. Saunders; 2011. p. 650-67.e2. Available from: https://doi.org/10.1016/B978-1-4377-0774-8.10062-4.

[20] Gao Y, Cai MX, Tian B, Lin H, Jiang ZY, Yang XC, et al. Setting 6-minute minimal examination time improves the detection of focal upper gastrointestinal tract lesions during endoscopy: a

multicenter prospective study. Clinical and Translational Gastroenterology. 2023;14(8):e00612.

[21] Hsieh YH, Koo M, Tseng CW. Factors associated with prolonged cecal insertion time in patients undergoing water exchange colonoscopy. Journal of Gastroenterology and Hepatology. 2022;37(7):1326-32.

[22] Castaneda MR. What Is "Real-time Control" and Why Do You Need It? Technical Article, Texas Instruments. 2023. Available from: https://www.ti.com/document-viewer/lit/html/SSZT084.

[23] Lustenberger F, Lehmann M, Cavalier L, Blanc N, Heppner W, Ernst J, et al. A colour 3200fps high-speed CMOS imager for endoscopy in biomedical applications. In: Proceedings of the 30th European Solid-State Circuits Conference. IEEE; 2004. p. 415-8.

[24] Apparatus H. Flexible Endoscope Specifications 72-7659; 2025. Accessed: 2025. Available from: https://www.harvardapparatus.com/flexible-endoscopes.html.

[25] Vishay. VEMD2704 Silicon PIN Photodiode; 2025. Available from: https://www.vishay.com/en/product/80304/.

[26] Regtien P, Dertien E. 7 - Optical Sensors. In: Regtien P, Dertien E, editors. Sensors for Mechatronics (Second Edition). Elsevier; 2018. p. 183-243. Available from: https://doi.org/10.1016/B978-0-12-813810-6.00007-0.

[27] Tran TN, Lam BM, Nguyen AT, Le QB. Load forecasting with support vector regression: influence of data normalization on grid search algorithm. International Journal of Electrical and Computer Engineering. 2022;12(4):3410-20.

[28] Bunch RM. Optical Systems Design Detection Essentials: Radiometry, Photometry, Colorimetry, Noise, and Measurements. IOP Publishing; 2021.

[29] Bascetta L, Magnani G, Rocco P, Migliorini R, Pelagatti M. Anti-collision systems for robotic applications based on laser time-of-flight sensors. In: 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE; 2010. p. 278-84.

[30] Nandal A, Zhou L, Dhaka A, Ganchev T, Nait-Abdesselam F. 3.4.4 Momentum. In: Machine Learning in Medical Imaging and Computer Vision. Institution of Engineering and Technology (The IET); 2024. p. 81. Available from: https://app.knovel.com/hotlink/pdf/id:kt0140UF0K/machine-learning-in-medical/momentum.

[31] Arduino. Arduino Mega 2560 Rev3; 2025.

[32] Instruments T. 8 Channel Analog Multiplexer/Demultiplexer; 2004.

[33] Inc MT. 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface; 2008.

[34] Helander HF, Fändriks L. Surface area of the digestive tract–revisited. Scandinavian journal of gastroenterology. 2014;49(6):681-9.

[35] Imaging Approach to the Small Intestine. In: Federle MP, Raman SP, editors. Diagnostic Imaging: Gastrointestinal (Third Edition). third edition ed. Diagnostic Imaging. Philadelphia: Elsevier; 2015. p. 342-7. Available from: https://www.sciencedirect.com/science/article/pii/B9780323377553501173.

[36] Sun Y, Liu Y, Su Y, Lueth TC. Model predictive control of 2-dof tendon-driven continuum robot using optical tracking. In: 2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM). IEEE; 2024. p. 1278-83.

[37] Hachen M, Shentu C, Lilge S, Burgner-Kahrs J. A Non-Linear Model Predictive Task-Space Controller Satisfying Shape Constraints for Tendon-Driven Continuum Robots. IEEE Robotics and Automation Letters. 2025.

[38] Bruder D, Fu X, Gillespie RB, Remy CD, Vasudevan R. Data-driven control of soft robots using Koopman operator theory. IEEE Transactions on Robotics. 2020;37(3):948-61.

[39] Kingbright. 1.0X0.5MM (0402) INFRARED SMD LED; 2025. Available from: https://www.kingbrightusa.com/product.asp?catalog_name=LED&product_id=APHHS1005F3C-70MAV.

A First Prototype

An initial proof of concept was developed before developing the robot described in this work. This proof of concept enabled the initial testing of most components and systems, as well as identifying shortcomings in the design.

A.1 Prototype Design

In figure A1, the original robot arm design can be seen. There are a few main differences between this design and the final design, as shown in image 1.A. Only the end segment of the prototype robot is actuated. The other segments were kept static using tape. The robot had 4 sensors at its end segment, one pair per degree of freedom.

The robot was controlled using PID code running on the Arduino MEGA, with a small prototyping board to connect the electronics, as seen in image 1.B.

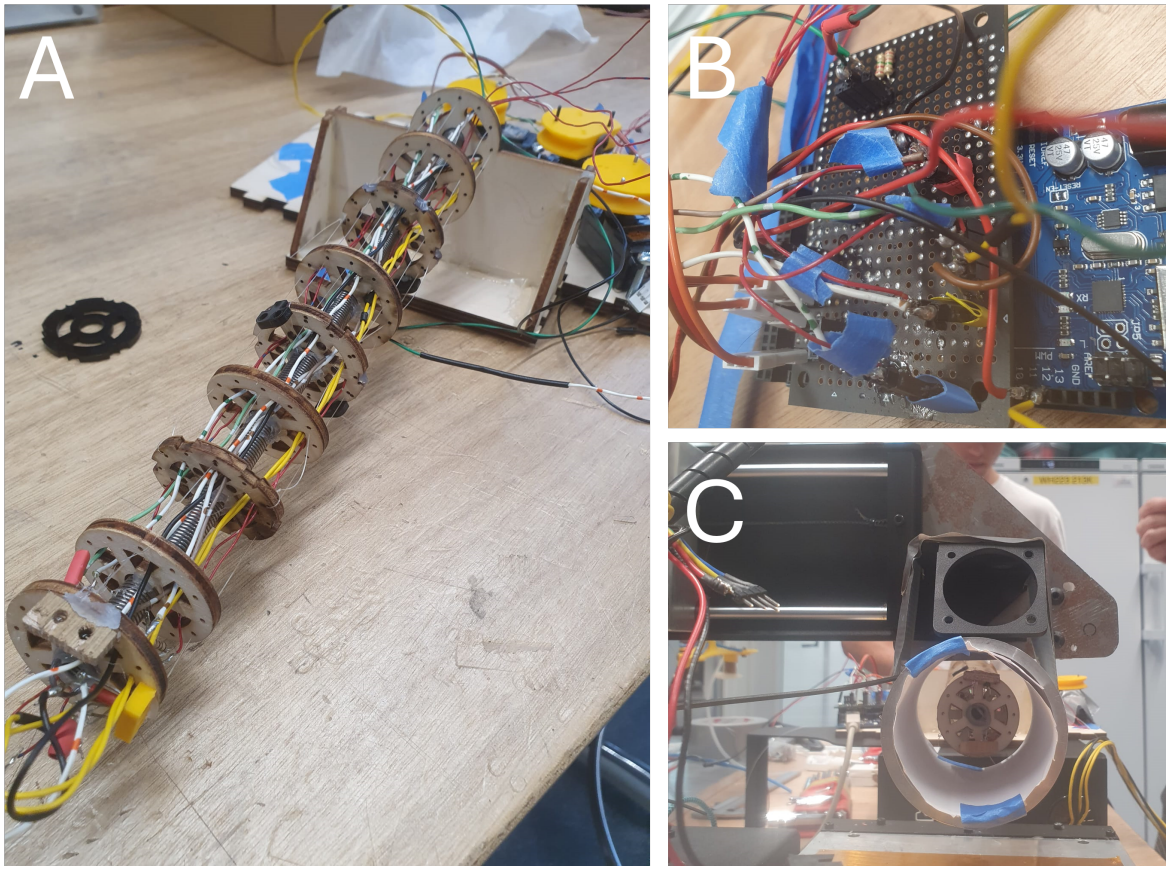

Figure A1: Images of the first robot prototype. A shows the full robot, with the sensors on the first segment, and the servos on the plate at the end. B shows how the wires of the robot were connected to the Arduino. C shows the test setup of the first prototype, where the tube is connected to 3D printer.

A.2 Prototype Experiments

The effectiveness of the robot's sensing, actuation and control was tested using a tube segment that was connected to a 3D printer. The robot was positioned inside this tube, as seen in image 1.C. This tube was then moved in a two-dimensional plane to quantify the deflection of the robot in different directions. However, this experiment was flawed, as it was very hard to see the response of the robot whilst inside the tube segment.

A.3 Main Lessons

The main shortcomings of the prototype were identified as:

- Wiring each sensor individually would require too much space inside the robot arm
- Having four sensors per segment leaves large gaps in the sensor field of view and makes the robot less likely to correctly estimate the centre of a curve.
- An improved method to fix the sensors to the disks was required
- An improved method to estimate robot response during experiments was required.
- Wood is not a good material for disk construction due to impurities in the wood.

B Programming

As the control of the robot was performed on a Raspberry Pi, it was necessary to program this control so that it would function on this hardware. In this is was important to design software that would perform consistently, whilst allowing all the different components to interact with one another. For this, a multithreaded C++ program was devised.

C++ was an ideal candidate programming language for this task for a few reasons. First and foremost, Matlab allows the conversion of Matlab functions to C++ functions, which means that the mathematical basis of the controller could be developed in Matlab, and no effort would need to be expended in accurately converting this to C++. Furthermore, I have a lot of experience with C++, which allowed for a fast development cycle.

B.1 Mathematics to C++

The mathematical basis for the MB controller was developed using the MATLAB symbolic toolbox. This allowed for fast development and easy validation of the gradient descent calculation. The final symbolic functions were then converted into numerical functions, which is possible with the MATLAB symbolic toolbox. These numerical functions were significantly faster than their symbolic equivalents and were the functions used during the control simulations. They were then further converted into numerical C++ functions using MATLAB. They were ported over to the Raspberry Pi. A test was run to confirm that the MATLAB and the C++ code would produce identical outputs if given the same input. This was the case, which showed that the conversions performed as expected, and no mathematical accuracy was lost.

B.2 Graphical User Interface

Using the open-source Raylib game development library, a simple graphical user interface was constructed to debug and interpret results, which can be seen in figure A2. Using a game development library allowed for real time updating of a 3D scene. This 3D scene, seen at the centre of the image, shows the four segments, as

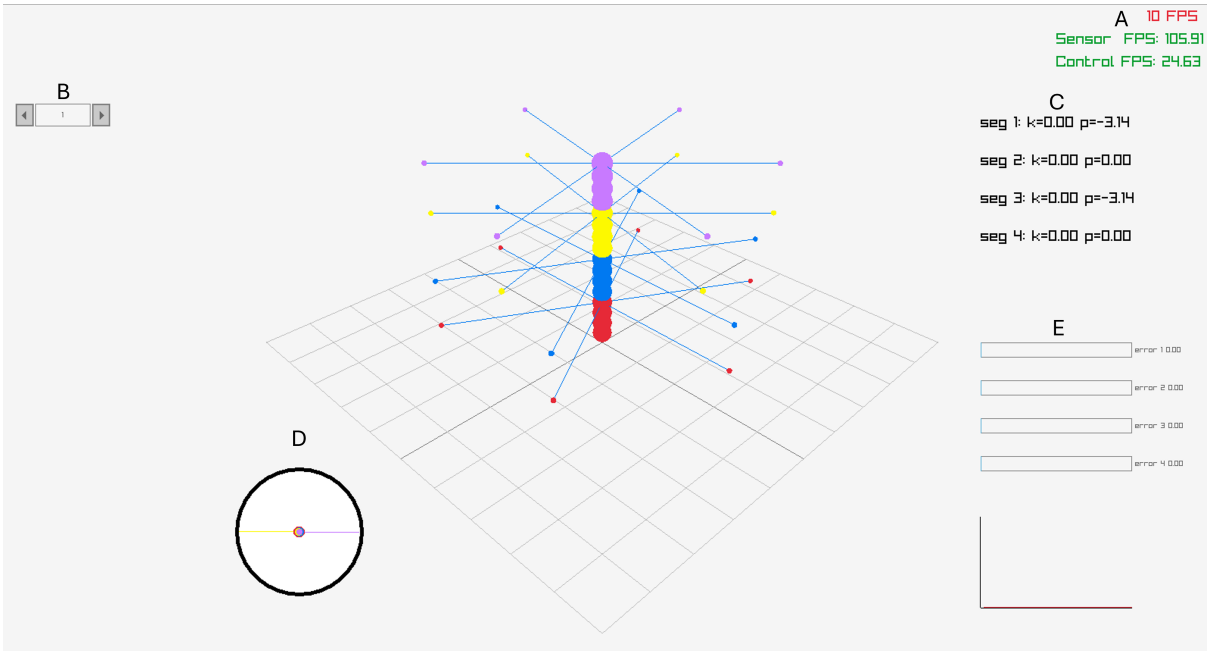

Figure A2: Screenshot of the graphical user interface that was developed for the robot. In the centre a 3D view of the controllers' estimation of the robot is visible, as well as the estimated surroundings, visualized with points. The labels were added to mark the different parts of the GUI. A shows the frame rate of the GUI, the refresh rate of the sensors, and the refresh rate of the controller. B is the selector used to select segment q_d for debugging. C shows the current segment variables. D is the polar representation of the variables in C. E shows the error per segment, as well as the overall loss of the robot in a graph.

calculated by the Constant Curvature model. It also shows the estimated position of the robot's surroundings, with lines showing the distance from the environment to the sensor. Using Raylib also allowed for real-time inputs using the keyboard. With this, the robot movement can be easily paused or continued by pressing the "P" button. Furthermore, this enabled manual movement of the q_d points that the robot would need to follow in x, y and z directions. Using the selector marked B, we were able to select which segments q_d would be manipulated. As implementation of the control was difficult, and at times the results of the controller were unintuitive, C,D and E were added. These allowed for a clearer interpretation of the controller and the gradient descent results.

B.3 Multithreading

The robot software has two systems that can work independently from one another, this being the controller and the sensor algorithms. By applying multithreading, we can separate ensure that the constraints of one system do not constrain the other, allowing both to run as fast as their design constraints allow.

B.4 Code

The code of the robot itself can be found on https://github.com/Ivenvh/Robotic_control.git. Which includes all working code and a command to compile the software. Matlab code and experimental data are available upon request.

C Sensor Identification

After the first prototype was designed, a redesign of the segment disks was made. To test the effectiveness of this new design, the test setup is seen in figure A3.A. The test consisted of a section of the disk segment on a pedestal. This was connected to a rotating slider. The rotation was measured by a potentiometer in the base of the pedestal, whilst the slider position had to be manually recorded. In image A3.B, the position of the sensors is shown. The results of these tests were used to estimate the 14th-order polynomial used in the sensor processing.

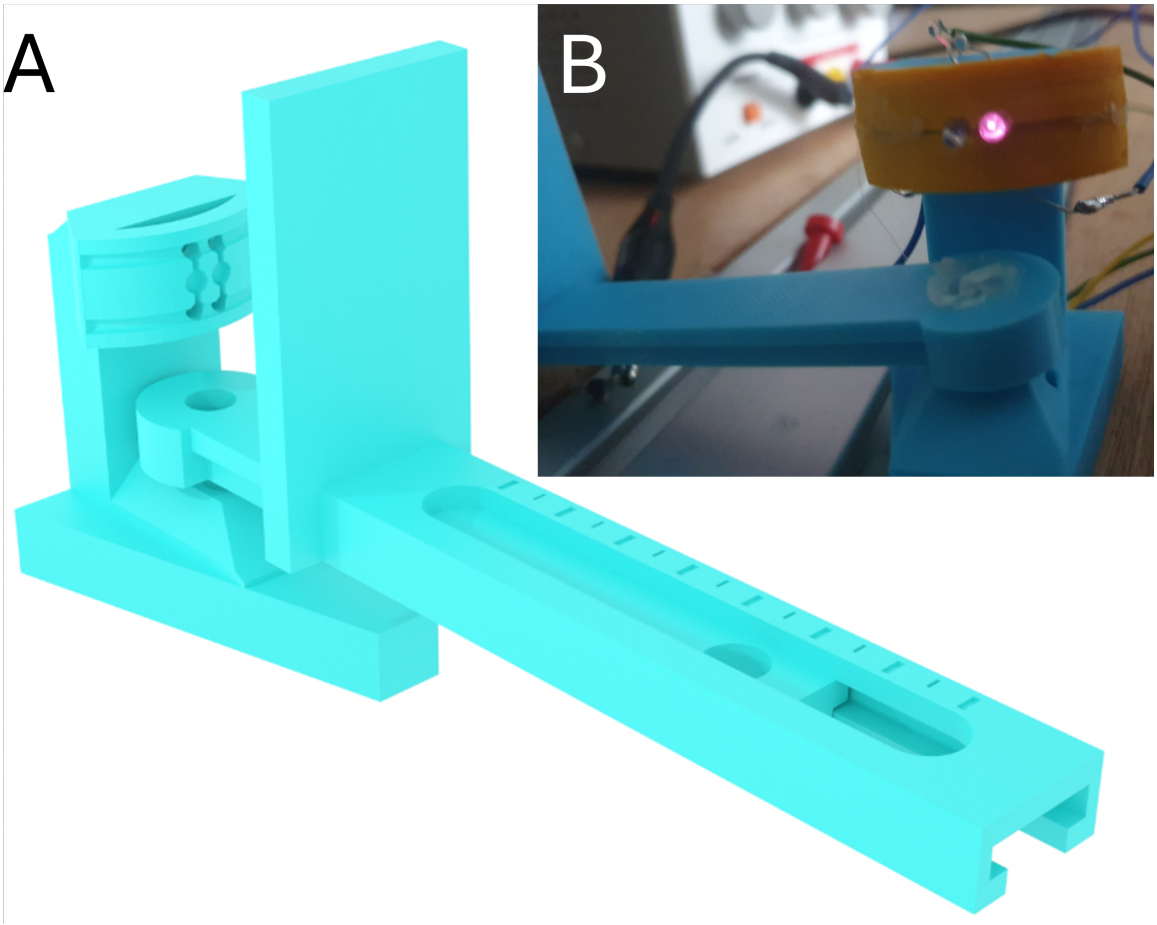

Figure A3: Test setup for the sensors is shown. A shows the design of the sensor test device. B shows how the sensors are placed in this device.