ECOLE CENTRALE DE NANTES

MASTER 1 FINAL PROJECT REPORT

Presented by

Prithvi Bharadwaj Mellacheruvu

Grama Srinivas Shourie

On June 23, 2022

Sensor-based shape estimation of continuum robots

Supervisors:	Federico Zaccaria	PhD student, LS2N, Nantes
	Sébastian Briot	Researcher, LS2N, Nantes

Laboratory: Laboratoire des Sciences du Numérique de Nantes LS2N

Acknowledgements

We are very grateful to Mr. Federico Zaccaria, doctorant, LS2N, for providing us with this opportunity. Thank you for the constant support that you have provided us, with all minor and major issues in the project, be it in practical or theoretical knowledge.

We sincerely thank Prof. Sebastian Briot for providing us with the opportunity to study, analyse and use various methods and sensors to estimate the shape of continuum robots. It was a very enriching experience.

We also thank our institution, Ecole Centrale de Nantes and the LS2N laboratory for this opportunity.

THANK YOU

Contents

List of Figures

Abstract

The goal of this project is to develop a sensor based state estimation of a cantilever beam which is usually used in continuum parallel robots. Two types of sensing methods have been tested/experimented and the best method is put forth.

A theoretical model is used to simulate the beam for a given point load in the free end. Another experimental model is also developed to estimate the bending in the beam based on the angle data from sensors positioned at specific locations.

The results of the 2 models are compared to find an estimate of the error. An optimisation is also carried out for the placement of the 2 sensors along the beam to find the locations that minimise the error. The outputs of these models are plotted and qualitatively verified with visual observations and experiments.

1 Introduction

1.1 Continuum robots

A robot, by definition, is a machine that is programmable by a computer capable of carrying a few tasks automatically. Generally when we think of robots, we think of them having rigid links. Continuum robots are types of robots that are characterized by infinite degrees of freedom. The numerous degrees of freedom help the robot to navigate in difficult to access environments, change shape at various points along the body, give better dexterity and adaptability, etc. They also help continuum robots to have compliant end-effectors to easily hold objects.

Figure 1: A three-segment continuum robot prototype. It is shaped by twelve tendons connected to six linear actuators. Reference [16]

According to [25], continuum robots have bodies that can form curves with continuous tangent vectors. These robots are inspired by life, for example, octopus, snake, elephant's trunk, etc. Currently, the continuum robots industry is growing and has many applications, for instance, in medical field and space exploration.

Continuum robots can be typically classified based on their structure or method of actuation. Based on structure, continuum robots could either be of single-backbone, multiple parallel backbone [14], or concentric tube type [7]. The type of structure decides the stiffness, strength, dexterity, adaptability, etc. The core may affect these properties, but the material chosen for the exterior of the robots also influences various properties of a given robot. Based on method of actuation, continuum robots may have actuators either externally outside of the robot or placed internally in the robot [18].

As such, continuum robots could be manufactured with different parameters [14] such as a given core, external material, the method of actuation, and the robot may even be controlled based on different theories, but there is one aspect of prime importance which is the sensing of the robot's & the end effector's shape, position and the force applied. This is very important as we cannot apply "standard rigid robot techniques" since the robot is elastic & the configurations depend also on the forces applied to the system.

Since these continuum robots find themselves in extremely delicate applications like medical surgeries [3], material handling in external space [9] , etc, force and shape sensing is crucial. The study of shape sensing of continuous robots is not yet well developed. Cameras may be used for this purpose but it is not appropriate for applications like invasive medical surgeries (though other methods have been developed as mentioned in upcoming section

1.3). Hence, sensing of shape and the force at end effector and feedback control constitute the challenges posed by continuum robots. This project is an effort to sense the shape of a continuous beam structure using various sensors and methodologies.

1.2 Problem statement

As mentioned before, force and shape sensing of continuous robots is crucial, and in this project we attempt to sense the shape of a continuous beam and simulate it. Such beams are used in continuum robots. These beams have extreme bending properties, that cannot be modeled by classical mechanics. Special models have been studied for such beams as in [20].

For the current project, we aim to estimate the shape of beams using various sensors and methodologies, as shown in the diagram below. The sensors we used in the course of this project provide data, from which angle of bend, with respect to the horizontal, at various points can be determined. In this project, the sensor data(from optimally placed sensors) and the mathematical model (as given in [20]) are used to simulate the shape of the continuous beam in MATLAB, and the error in shape estimation is analyzed.

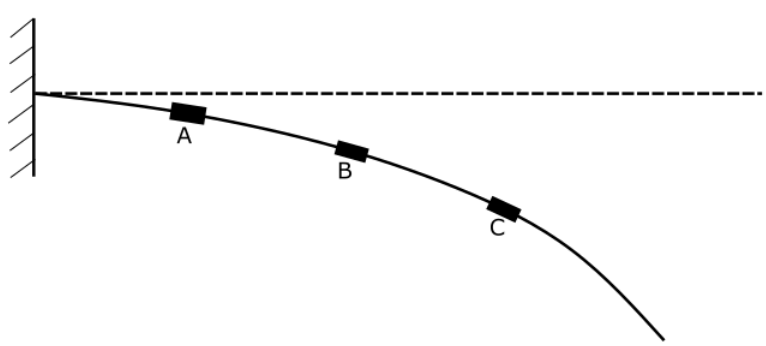
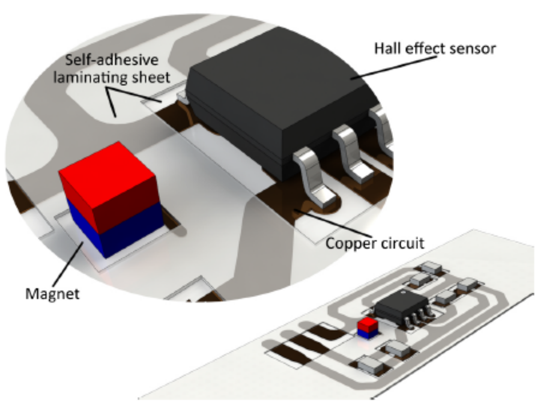


Figure 2: Shape estimation of bending beams using sensors placed along the length of the beam

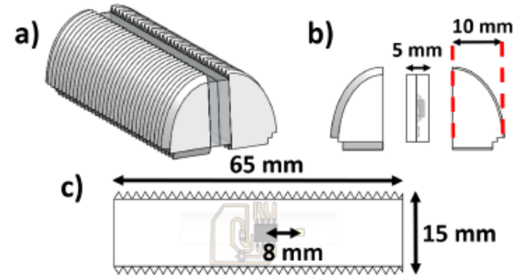
1.3 State of the art

Continuum robots seem like modern technology, but they have been in development since a long time. This is indicated by papers like ones published on biologically inspired robots [8] in 1993, dexterous handling of complex objects [6] in 1994, controller for a tentacle-like robot [12] in 1995, and also one on a robot finger [1] in 1976. Hence there is quite a lot to be learnt from history so that we can sustainably develop in the future. This section provides a brief state of the art in the field of continuum robots and their shape estimation.

Continuum robots, as mentioned in 1.1 may be actuated in many ways and pneumatic actuation is one of those ways. Pneumatically actuated robots are very compliant by nature and they can gently handle objects. The behaviour of these pneumatically actuated continuum robots is very dynamic and non-linear, and shape sensing of these robots is established by Hall effect sensors or resistive flex sensors, conductive fluids or even the novel optical fiber Bragg grating technique. But these methods of shape sensing have limitations (they may be expensive, non-generic, computationally expensive, etc) and cannot be used in all applications. The paper [17] describes the development of a magnetic curvature sensor, as shown in fig. 3. The paper also compares the novel sensor with a resistive curvature sensor. It was experimentally evaluated that the magnetic curvature sensor gave noisy but accurate results compared to the stable but slightly erroneous results given by the resistive sensor. The resistive curvature sensors also has a considerable time delay when the robot is de-actuated which does not occur with a magnetic sensor. Hence, [17] concludes that an array of magnetic sensors can be used to reconstruct the shape of pneumatically actuated continuum robots.



(a) Magnet sensor. Reference [17]



(b) Assembly of magnet sensor in the robot body. Reference [17]

Figure 3: Magnet sensor and its assembly into the continuum robot as described by [17]

The article [24] summarizes the general state of the art in the field of proprioception(shape sensing), exteroception(tactile sensing) for soft robots. The article gives description of textile electrode-based sensors, piezoresistive sensors, sensing skins, capacitive sensors, stretchable electroluminescent skins and other methods for proprioception and exteroception. Hence this article serves as a guide to anyone trying to understand the basic sensors used for sensing for continuum robots.

While the article [24] provides a summary of the sensing technology, the article [14] provides an even more comprehensive study of development of continuum robots, their structure, material and actuation methods spanning over two decades. Continuum robots are mostly bio-inspired, for the most part their structure. Study of various structures and backbones, like elephant trunk, snake, reptile's trunk, with either single or multiple backbones, possibly with disks, and more study has been performed in this article. Fig. 4 shows some of the current bio-inspired designs. Materials used for continuum robots affect their flexibility, strength, dexterity and softness. Advantages, limitations and applications of various materials for soft robots has been well described in this article. Actuation methods including multiple actuation techniques are also mentioned. The article concludes that a lot of work has been done in the field of soft robots and that soft robots can be used for applications in industrial manufacturing as well. Fig. 5 shows various soft robot end-effectors.

Figure 4: Continuum robot structures inspired by Octopus leg and by tentacles. Reference [14]

Figure 5: Continuum robots inspired by human hands. Reference [14]

Soft robotics is to be particularly used for applications where the robots have to traverse confined spaces in curvilinear paths. That includes applications in space and interventional medicine. Article [3] describes how continuum robotics science and medical applications come together. The focus of this paper is on continuum robots that are used in minimally invasive surgery because these robots tend to have a smaller footprint, and they can traverse to difficult to access surgical sites using non-linear paths, and can finish the surgery with a tender but dexterous effect. Article [3] provides design, structure and actuation methodologies as in [14], but in a more precise area of interest regarding medical applications. The kinematic models and frameworks used are also described along with inverse kinematic frameworks and control methods. The article describes usage of continuum robots in specific applications in various fields of medicine like Neurology, Urology, Cardiology, etc. To add, the article also specifies the challenges faced in the study of continuum robots like lack of distinct sensing technologies, integration of systems, etc. Hence a study of continuum robots for medical applications was conducted and it predicts that there will be a lot more development in this field.

It is very difficult to predict the shape of continuum robots, even with highly studied kinematic models, when they are faced with unpredictable and dynamic loads. But still they have the advantage of being highly capable of accessing target sites in anatomically complex environments. As an addition to article [3], article [22] provides a grand summary of some of the sensing technologies developed for continuum robots that are used in minimally invasive surgeries. The article describes the fiber optic sensors based(fiber Bragg grating(FBG) techniques), electromagnetic(EM) tracking based and intraoperative imaging modalities based shape reconstruction methods. They have their own advantages and disadvantages. FBG methods have small sizes, invariant in EM fields, and are very accurate and effective in smooth structures like active needles and concentric robots. But FBG methods are very expensive and sophisticated. EM tracking are much more simpler to use and offer freedom from direct line-of-sight, but EM fields will disturb measurements and tracking is not accurate. Intraoperative imaging uses techniques like flouroscopic or endoscopic imaging or ultrasound imaging. They help in direct measuring and shape reconstruction of the robots. But there is a lot of radiation exposure and these imaging techniques are expensive as well. This article [22] also mentions that a combination of these implementations is much more effective when compared to stand-alone implementation of either of them.

The paper [21] proposes a novel method to reconstruct the 3D shape of a beam, using groups of three strain gauges and also a reconstruction method that handles large deformations and 3D finite strain in the material. Repeatable accurate reconstruction of the beam shape was obtained by this method. A couple of Groups of three strain gauges were used at various locations along the beam to obtain strain values along the beam. These values were sent to a reconstruction method that calculated the 3D shape of the beam. It was also concluded that the method was applicable to beams of any cross-section but only to the circular ones. This method has also been postulated for use in needle shape monitoring in medical fields

The authors in [15] try to address the fundamental problem of continuum robots i.e. sensor selection and their placement. It says that sensor selection and its placement are inter-connected for state estimation and it uses an Observability matrix whose rank analysis gives useful insights on the selected placement configuration. The following figure shows the result of optimizing the sensor placement problem.

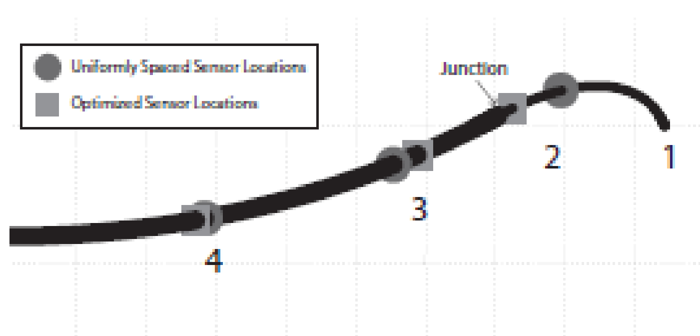


Figure 6: Result of optimizing the problem of sensor placement on a beam to effectively sense the bending. Reference [15]

The shape reconstruction framework proposed in the paper [13] provides a standardized approach to predicting the accuracy of a sensing system based on optical fibers with fiber

Bragg gratings by proposing several types of reconstruction models. They demonstrate how optimal sensor locations can be determined via an optimization problem that minimizes the shape and tip errors between the reconstruction model and a mechanics-based model.

The article [4] refers to a method of shape estimation of continuum robots using IMUs as done during the course of this project. However, the use of IMUs in our project is not inspired by this article. This can be ascertained from the date of its publication which falls during the course of our project.

1.4 Plan of action

This project requires us to develop a system which uses sensors to collect the angle of bend at various points on the beam and then pass it on to a MATLAB program to re-create / estimate the final state of the beam due to the load. The below flowchart will give a clear view of the workflow:

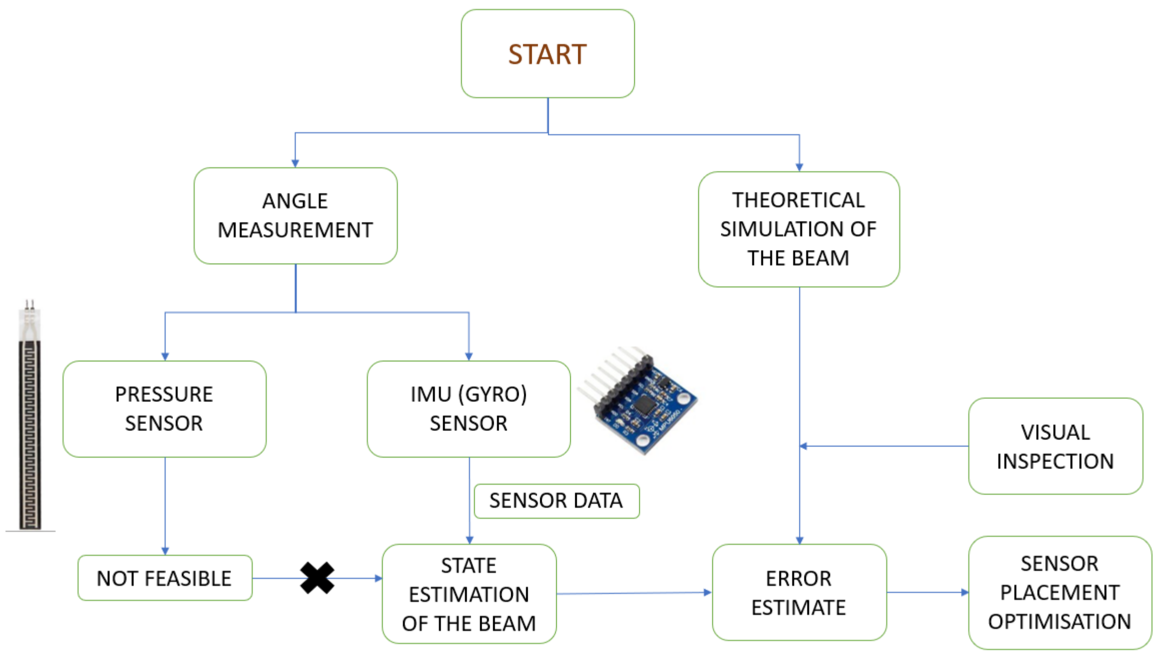


Figure 7: Work-flow chart

2 Mathematical Model

There are different methods available in the literature which are suitable for different applications and have varied scope. We have used the model which is given here [20] to solve for the position estimation of the beam/arm as we find it suitable for our case.

We have implemented this model in two parts:

- Theoretical reconstruction of the beam using information of the loads applied and the beam properties.

- Estimated reconstruction of the beam using information solely from the sensors placed on the beam.

MATLAB is used to simulate the results of our model and there are different codes used & developed for the two types of implementation.

2.1 Theoretical Model

The equations are derived as shown in chapter 2 of [20] where the rod is a parametric Cartesian curve in 3D space paired with an orthonormal rotation matrix expressing the material orientation.

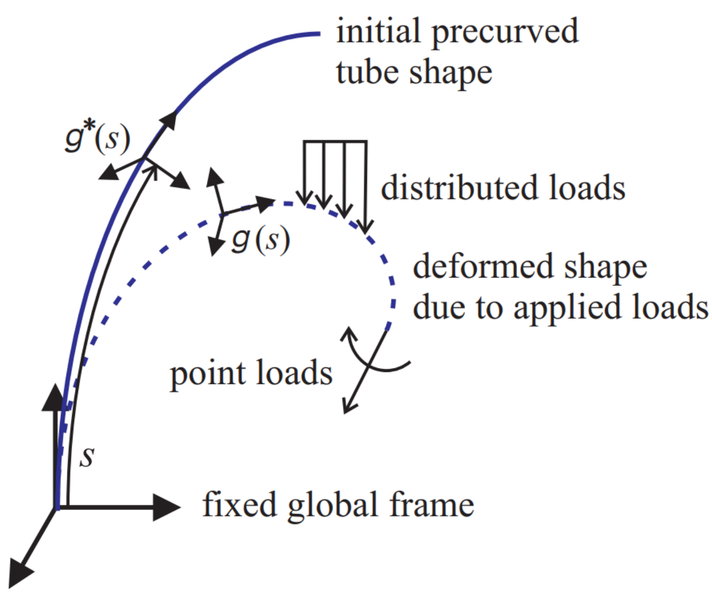


Figure 8: The shape of a rod structure in its reference state is defined by a parameterized frame along the rod's length. The rod then deforms to a new shape defined by a new set of frames as the result of external forces. Reference [20].

In order to arrive at a full set of equations that can be used to calculate the shape of a deformed rod, we must combine the geometric descriptions with the equilibrium and constitutive laws. A simplified model can be obtained if the effects of shear and extension cause relatively small changes in the shape in comparison to the effects of bending and torsion. This yields a full set of differential equations and the planar 2D case, which is used for this project, is shown below.

$$\dot{p} = Re_1 \tag{1}$$

$$\dot{\theta} = u = m/EI \tag{2}$$

$$\dot{n} = -f \tag{3}$$

$$\dot{m} = -\dot{p} \times n - l \tag{4}$$

The notations stand for :
$p = position\ vector,\ R = Rotation\ matrix,\ e_1 = standard\ basis\ vector$
$\theta = orientation,\ m = internal\ moment,\ E = Young's\ modulus$
$I = 2^{nd}\ moment\ of\ inertia$
$n = internal\ force,\ f = distributed\ load\ per\ unit$
$m = internal\ moment,\ l = distributed\ moment\ per\ unit$

The problem we solve in this project is the case of a point load applied at the end of the cantilever beam and this is implemented as follows:

1. The general idea is to solve the differential equations using the ODE45 method of MATLAB with initial condition being the fixed end of the beam where the position, orientation is zero but the force and moment (n & m) are not zero since there is a point load at the free end.

2. To get these n & m values, we follow an approach which uses the FSOLVE method of MATLAB which is used to solve a system of non-linear equations. The n & m values are first initialised to zero and then solved with FSOLVE where the system is basically a function named beamShooting which takes theses values and then passes it as initial conditions to the ODE45 solver for the differential equations. The solution of this solver gives us p, θ, n & m values for a set of discrete points along the whole length of the beam. We then take the force (n) value from the ODE solution for the free end of the beam and subtract this from our given point load value and if the result and also the moment (m) at the endpoint are equal to zero then this implies that we got the right initial values for n & m.

3. These values are then used to initialise the variables to call the ODE45 solver one last time to solve for the system of differential equations defined previously from equations 1 to 4. This gives us the final p, θ, n & m values for a set of discrete points along the whole length of the beam.

4. The load value and the material properties are defined as variables in the beginning of the program which can be varied for different simulations depending on the conditions required.

5. The position values from the final result is then taken to plot the graph which is similar to the following figure.

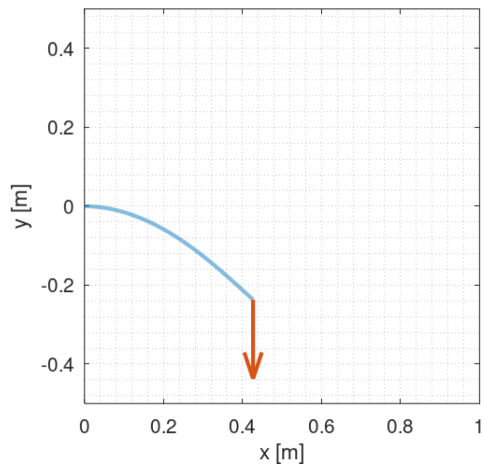


Figure 9: Blue line indicates the beam position in 2D space and the red arrow shows the load which is 0.2N for this simulation.

2.2 Sensor based Model

This is a modified version of the previous section 2.1 where the position equation 1 is the same but instead of calculating the θ values by solving the rest of the equations for the specified loading conditions, we take the orientation reading from the sensors fixed at specific positions along the beam and generate a spline of these values for the full length of beam. These θ values are then used to construct the rotation matrix which in turn helps us in position estimation. The data from the sensors is acquired by the Arduino board and then streamed to the MATLAB code in the computer through serial communication. The sensors in our final working model are MPU6050 Inertial Measurement units (IMU) (reference - [11] [5]) which houses an accelerometer and a gyroscope. We take the gyro values for θ as mentioned.

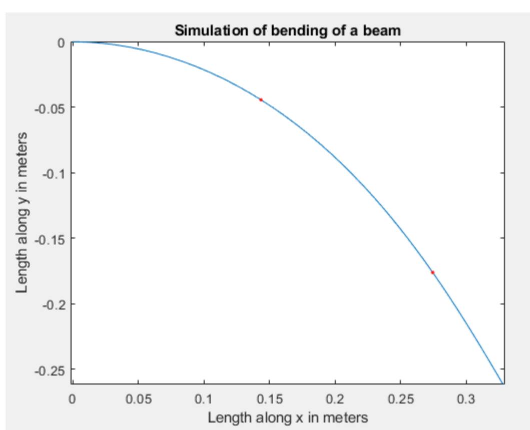


Figure 10: The estimated shape of the beam for a random load.

Our plots are near real-time i.e. there is a slight time delay in the beam position visualisation.

3 Sensing Methods

As mentioned before, this project is an attempt to sense the bending in a beam that constitutes a continuum robot. Various sensors and sensing methods were considered for this purpose. The first sensor that was considered was the resistive force sensor. Different methods were considered to fix the sensor on the beam. Resistive flex (bending) sensor was also considered for this project, but the finally agreed upon sensor was the IMU (Inertial measurement unit). IMU was able to provide us with good, repeatable, real-time data to simulate the bending in the beam. This section details on each of these methods used for sensing the bending in the beam.

3.1 Resistive force sensor

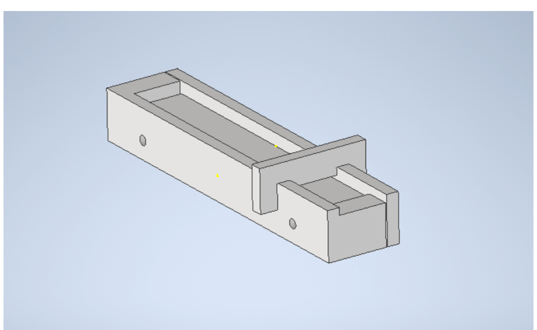
Resistive force sensors are sensors that have a material whose electrical resistance changes when force is applied on them. These sensors give a value that is proportional to the amount of force applied on them. Fig. 11 is a figure of the resistive force sensor. The resistive force sensor is a very thin sensor, as thin as a sheet of plastic and is highly flexible. If it were attached to a beam, it wouldn't influence the bending of the beam.

Figure 11: Resistive force sensor

These sensors can be interfaced with Arduino micro-controller and their sensor readings can be read in the serial monitor. The interfacing between Arduino and the sensor, and the code for reading the sensor data from the serial monitor was obtained from [2]. The sensor output data was later mapped to larger values for ease in perception and understanding, i.e. from 0 to 1024 to 0 to 16384. This sensor output was also mapped approximately to an angle of bend from the start of the sensor to the end of the sensor.

Upon experimenting with the sensors, it was evident that the sensor responded when a force is applied on the surface of the sensor. The sensor is sensitive to light touches. But in the current application of sensing the bending in the beam, it was expected that the sensor data can be mapped to an angle of bend. Hence, it was decided that the sensor should be attached to the beam in such a way that the bending in the beam induces a force on the surface of the sensor, such that the angle of bend at the sensor position on the beam is directly proportional to the force on the sensor and also to the output of the sensor. Two methods were implemented to ensure this; one, to encapsulate the sensor in a silicone mould and mount it on the beam as shown in Fig. 12 and Fig. 13, two, to use a plastic envelope around the sensor and mount it on the beam with electrical tape, as shown in Fig. 14.

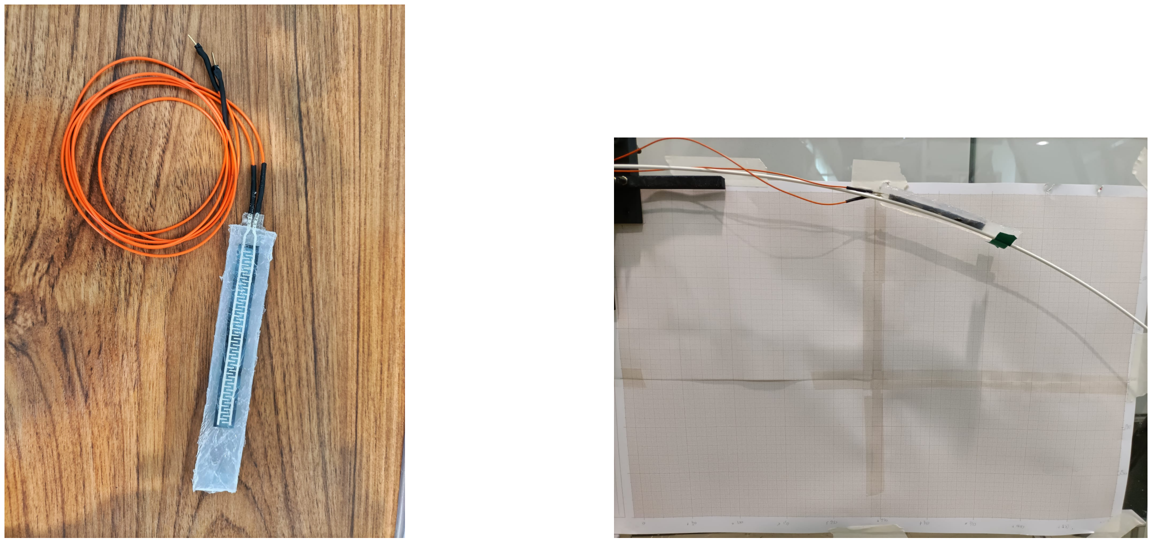
To enclose the sensor in a silicone mould, a cast was created so as to repeatably make moulds for the sensors. The cast was designed in Autodesk Inventor. Fig. 12 shows the design of the cast assembly and the 3D printed cast assembly. Fig.13 also shows the sensor in the silicone mould. Use of silicone mould around the sensor was expected to transmit the force due to bending to sensor as the mould was also a solid. Fig. 13 shows the silicone mould sensor mounted on a bent beam. Upon looking at the results, it was clear that the silicone mould made the sensor less sensitive to the beam. The silicone mould, being highly elastic, did not transfer the force effectively to the force sensor. Inaccurate, non-repeatable data was obtained. Hence it was concluded that this method is not a good method to measure the bending, and that direct contact had to be established between the sensor and the beam for effective force transfer.



(a)

(b)

Figure 12: 3D design(left) and assembly(right) of the cast that was used to enclose the resistive force sensor in silicone mould



(a) (b)

Figure 13: Resistive force sensor enclosed in silicone mould(left) and the sensor in the mould being used on the beam to obtain angle of bend(right)

It was understood that in order to obtain better values from the force sensor, direct contact was to be established between the sensor and the bending beam. To directly use adhesive to stick the sensor on the beam may have damaged the beam. Hence, we put a plastic cover around the sensor and attached it to the beam using electrical tape. The figure below shows the sensor in the plastic covering and also the sensor attached to the beam. The sensor readings this time were not accurate, but they were repeatable. The sensor, when used in the method mentioned here, gives good values when the beam bending is very high. Fig. 15 is plot that shows actual angular values and mapped outputs from the sensor between the start and end of the sensor. Hence, we could conclude that the sensor by itself was insensitive to low bending angles, where there wasn't enough force applied on its surface.

(a) (b)

Figure 14: Resistive force sensor enclosed in plastic sheet, with the help of electrical tape,(left) and the sensor in the plastic sheet being used on the beam to obtain angle of bend

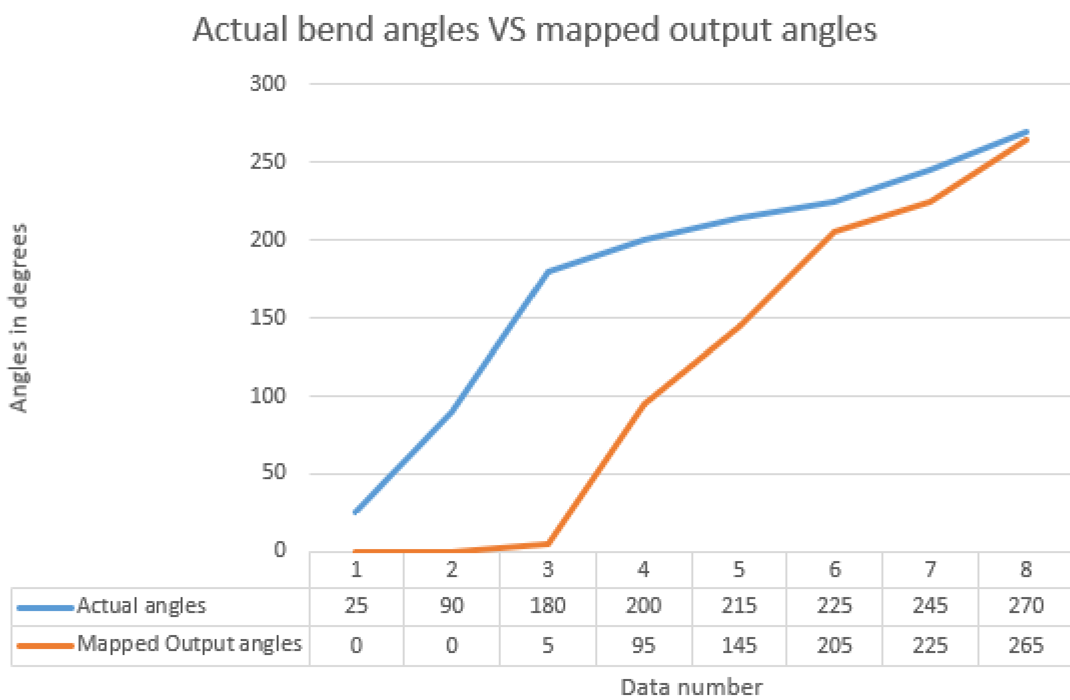


Figure 15: Plot of actual angular values and mapped outputs from the sensor between the start and end of the sensor

Hence it was concluded that a force sensor may not be the best choice of sensor to sense the bending in a beam. The next prospective sensor was the flex (bending) sensor.

3.2 Flex sensor

A Flex or bending sensor is a sensor whose resistance is directly proportional to the amount of bending in the surface that the sensor is attached to. It directly measures the amount of bending or deformation in the surface. Fig. 16 shows a flex sensor. Such flex sensors are thin, but not as thin as the resistive force sensor. Flex sensors are also less flexible(stiffer) when compared to the resistive force sensors. Even though flex sensors are used in applications like haptic gloves to measure the bending in fingers, it was supposed that they would not be able to sense the bending in beams, and also that they would influence the bending in the beam. So, flex sensors were not experimentally used during the course of this project.

We needed a sensor that could give us accurate, repeatable data, and that the sensor should be able to give us data in real-time. Since we required angular data from the sensors, we thought of using an IMU. Little thought was sufficient to indicate that it was a great idea, and we went for it, as discussed below.

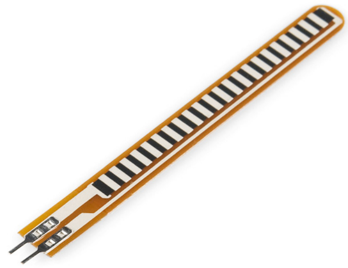


Figure 16: Flex sensor. Reference [23]

3.3 Inertia measurement unit

IMU is a sensor that contains accelerometers, gyroscopes, magnetometers and other such devices which as a whole are used to calculate orientation, linear and angular acceleration, forces and more. They find usage in many applications like aircrafts, spacecrafts, ships, submarines, automobiles, etc. We use IMUs to calculate the orientation of body at a point. IMUs as used in this project provide the rotation of a point about x, y, and z axes (roll, pitch and yaw angles). Fig. 17 shows an IMU.

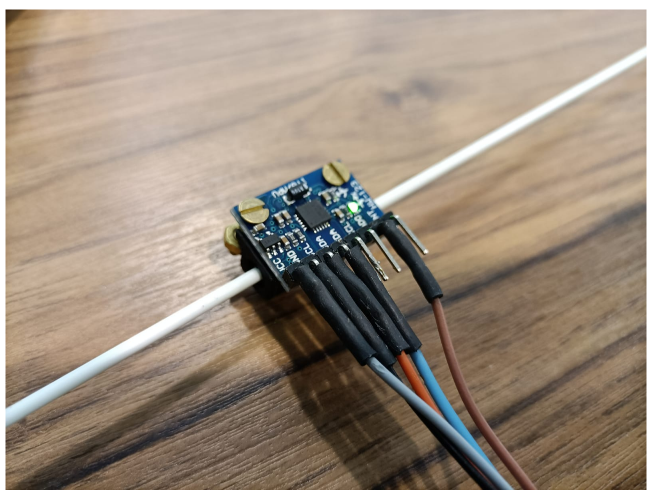
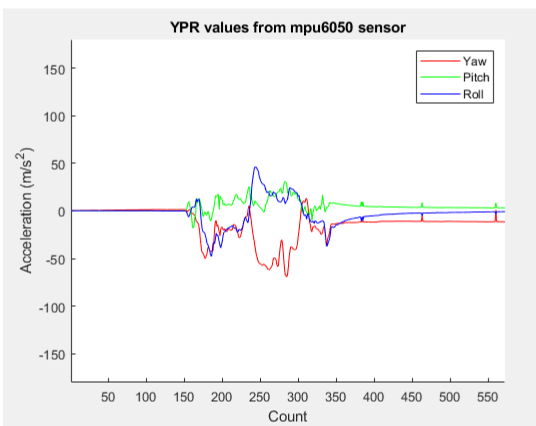


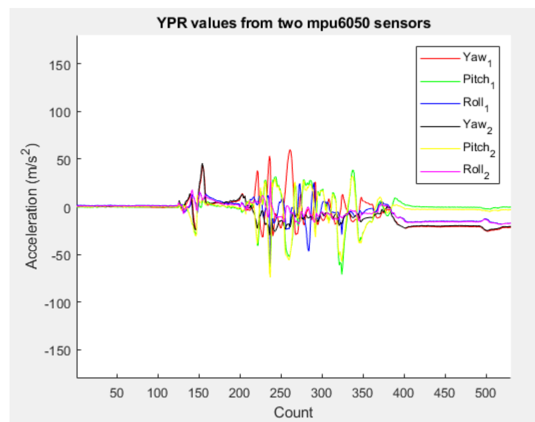
Figure 17: Inertia measurement unit attached to a beam using a mount

As mentioned in section 2, the mathematical model we have considered takes in angles, with respect to the horizontal, at various lengths along the beam and simulates the bending in the beam. In order to calculate the angles at various lengths of the beam, until now we have used the resistive force sensor and we have also postulated the use of the flex sensor. But now, we have found that IMUs can be used to get the angles of bend at various lengths of the beam. IMUs provide accurate, real-time, repeatable outputs and also they are small, light and do not influence the bending in the beam.

To make sure that good, real-time output can be expected out of the IMUs, and to also know how to calibrate the IMUs, one IMU was tested. The work of Jeff Rowberg [19] was the basis for our usage of these sensors with Arduino. His library had in-built calibration & off-set corrector and could access the Digital Motion Processor (DMP) of the MPU 6050 sensor. IMU was connected to Arduino with reference from [10]. We could get an accurate plot of the real-time yaw, pitch and roll values of one sensor, as shown below in fig. 18. Upon experimenting, it was found that the sensor is calibrated and zero-set at startup. Meaning, when the code of the Arduino micro-controller is run, and the first values of the IMU are requested, the IMU is calibrated and the IMU considers its current orientation as the reference for future values (yaw, pitch and roll angles are all zeros). Hence, zero-setting of IMUs is pretty simple. To zero-set the sensor horizontally, we fixed two long screws on the board (in level with the fixed end of the beam) as in fig. 19. When the calibration starts we place the beam with the sensors on the screws to zero-set the sensors to the correct horizontal. Once we start receiving values, we load the beam and take the beam off the screws.



(a)

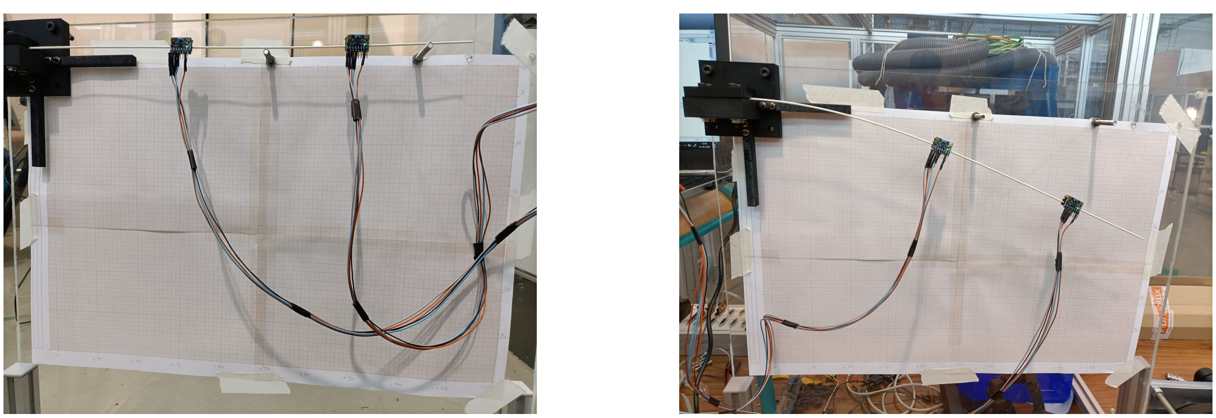


(b)

Figure 18: Plots representing the continous input of yaw, pitch and roll data from one(left) and two(right) IMUs

Figure 19: The experimentation board with screws that help us zero-set the IMU sensors to the horizontal at startup

Once this was achieved it was required that we interface more IMUs with the Arduino because the more the number of points where we know the angle of bend, the more accurate is our simulation. Upon researching on the internet, it was found that the IMUs we use for this project, MPU6050, have address pins. These pins, by default, are set to LOW. But we may reset them to HIGH. By the use of this technique, we may connect a maximum of two MPU6050s to a single Arduino micro-controller. The case of maximum of two MPU6050s to a single Arduino is when we do not have a multiplexer. If we had multiplexer, we could connect upto eight IMUs to a single Arduino. In this project, since we did not have a multiplexer, we use only two IMUs. Hence, the number of points where we know the angle of bend is three, two points where the IMUs are located and one at the fixed point of the beam. We consider the angle of bend at the fixed point of the beam to be zero. Fig. 20 shows the IMUs fixed to a bent cantilever beam.



(a) (b)

Figure 20: Horizontal beam during calibration(left) and a bent beam equipped with IMUs

Hence, two IMUs were connected to the Arduino micro-controller and respective yaw, pitch and roll values were obtained. In order to place the IMUs on the beam, we designed and manufactured(using 3D printers) mounts. These mounts can be screwed to the beams at various locations. Since we use screws to tighten the mounts at various locations, we could also use the same mounts on beams of various diameters. Fig. 21 shows the isometric design of the mount and the mount itself. The IMUs can also be easily placed and screwed onto these mounts.

(a) (b)

Figure 21: 3D design(left) and the 3D printed mount for the IMUs to be mounted on bending beams of various sizes

Hence, two IMUs have been mounted on the beam. These IMUs have been interfaced with the Arduino. They may be zero-set at startup. We receive angles of bend at various lengths of the beam. These angles are passed through the mathematical model in real-time and we receive the approximated model of the beam on the plot.

With multiple sensors comes the question of placement of the sensors. Where to place the sensors to get better a approximated model of the beam? Even though there may be several ways to mathematically formulate this question into an optimization problem, we used a simple method to find the optimal sensor positions. The following section details this optimization.

4 Sensor placement optimization

Sensor placement plays a very important role in the shape estimation of continuum robots. Depending on the mathematical model that is being used to estimate the shape of the robot, the importance of the sensor placement varies. The mathematical model for shape estimation used in this project as mentioned in section 2 and detailed in [20], uses the data that it currently has and always predicts forward. Meaning, if we consider a bending beam(as in this project), the shape of the beam is estimated starting from length zero towards its end, always in that order. This helps us realize that sensors placed towards the end of such a bending beam do not have much use, because the data provided by them is not used to predict anything. Hence, in applications of bending beams with the considered mathematical model for shape estimation, better results may be obtained by placing sensors before seventy-five percent of their length. In order to find the optimum placement of sensors we considered a simple approach rather than an extensive optimization problem. In order to do so, it was important that we come up with a way to find the error in the shape estimation. As mentioned in section 2, we also have a theoretical model to compare our estimation with. The position(in x and y directions) of each point along the length of the beam from the shape estimation is compared with the corresponding point in the theoretical model of the beam when subjected to the same weight at the free end to obtain the error in estimation. Fig. 22 shows the error along the beam when the beam was loaded. The mean of errors along the length of the beam in x and y directions could be obtained, and the root of sum of squares of these two means was considered to represent the shape estimation error.

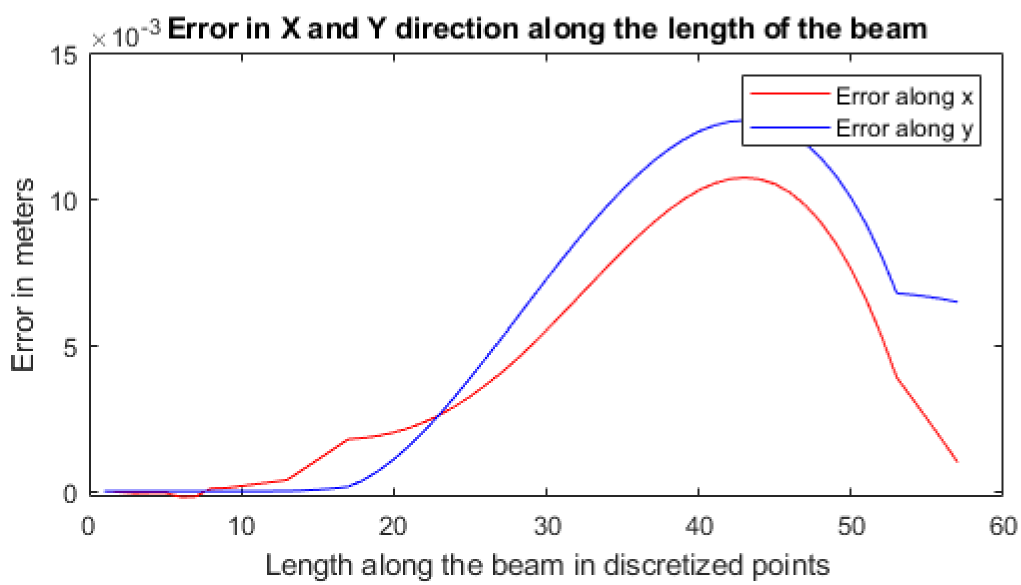


Figure 22: Positional error, in x and y directions, all along the length of the beam

Hence, shape estimation error can be calculated, basically, by comparing the theoretical model and the shape estimated model. This shape estimation error is used to find the optimum sensor placement for a given load. We iteratively choose all possible sets of two angular values and their corresponding lengths from the theoretical model(for a given load) and pass this angular information and corresponding lengths along the beam to the shape estimation model, whose output is again compared against the theoretical model. In order to find the optimum sensor placement we find the set of angular values for which the shape estimation error is minimum. The lengths along the beam to which these angular values correspond to are the optimal positions for the sensors. Fig. 23 and fig. 24 represents the output of optimizing the sensor placement for a beam of length 45cms loaded by 10gms at the free end.

```
lengthOfBeam =

    0.4500


load =

    0.1000


bestSensorOnePos =

    0.1045


bestSensorTwoPos =

    0.3091


bestError =

   1.5645e-12
```

Figure 23: Print statement of optimum sensor placement when beam length is 45cms and load is 0.1N

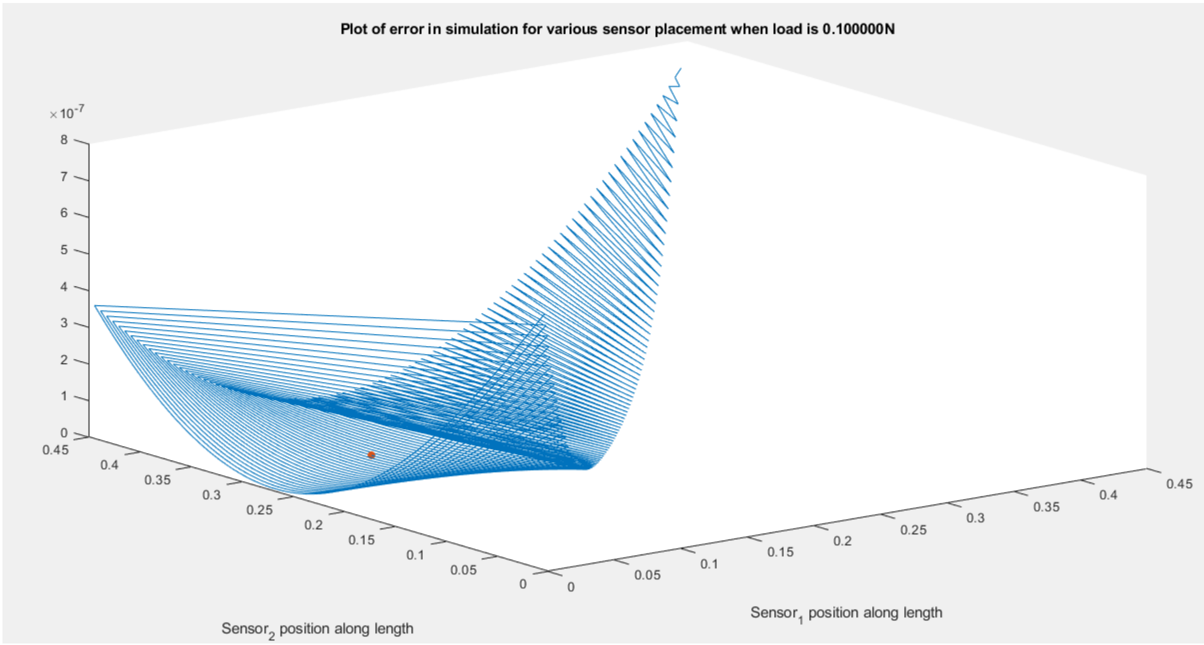


Figure 24: Plot of variation of shape estimation error(in meters) against sensor placement with optimum sensor placement marked(red dot)

As we see in fig. 23 and fig. 24, the error is minimal when the sensors are placed at their respective optimum positions and the error is in millimeters. So it may be concluded that the sensor placement is optimized.

5 Result

As mentioned in section 3, two IMUs are used in this project to obtain real-time angles of bend at various lengths along the beam. These angles help us to obtain an approximated real-time model of the beam. Also, as mentioned in section 4, we can optimize the sensor placement along the beam. Fig. 25 shows how huge a role the sensor placement plays in shape estimation of the bending beam. The estimated shape is an upward spiral whereas the actual bending and the theoretical model represent the beam bent to a small degree in the downward direction. This large error in shape estimation is because the sensors are non-optimally placed close to the fixed end of the beam.

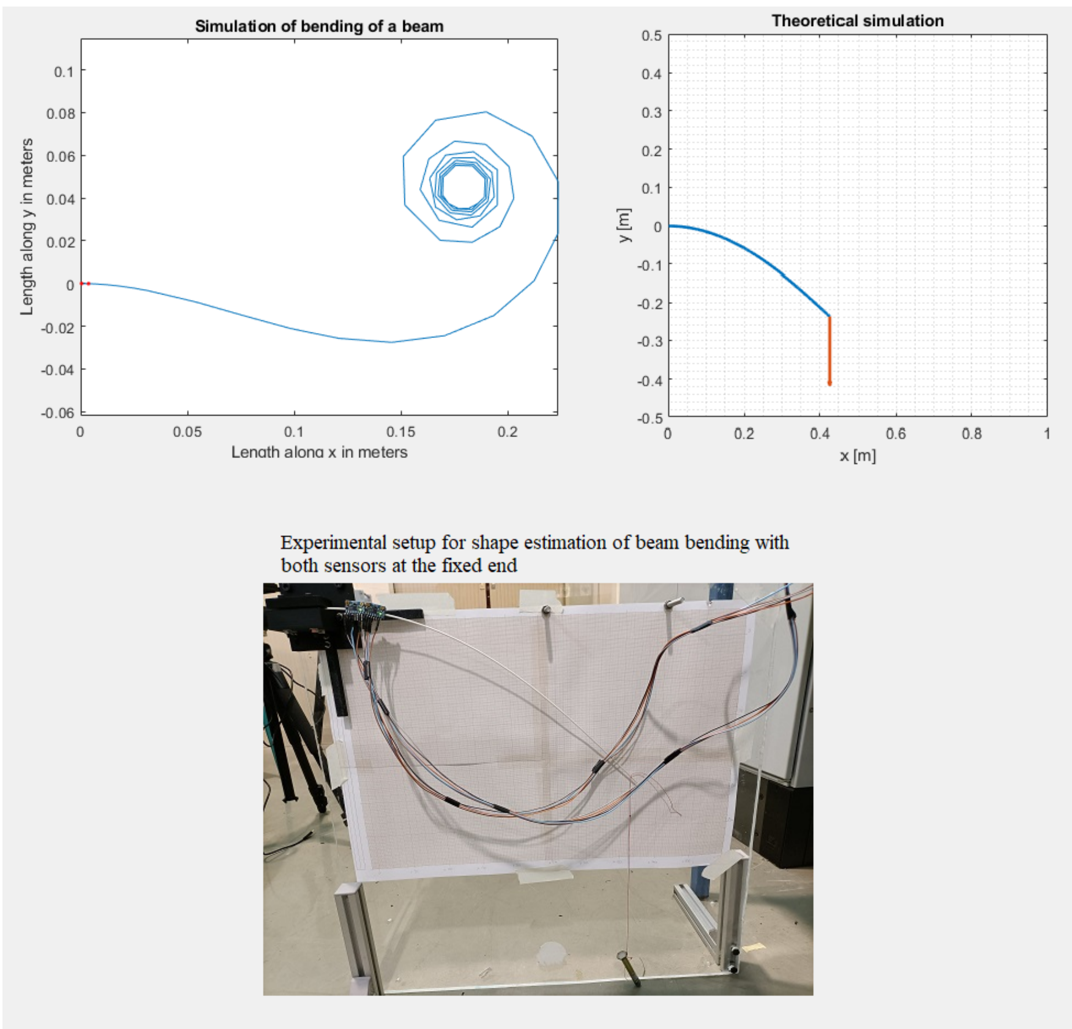


Figure 25: The beam bent due a load of 0.38N as both sensors at the fixed end send angular information to the computer(bottom), shape estimation of the beam(sensor positions are shown with red dots)(top left) and the theoretical model of the beam under a load of 0.38N(top right)

To further investigate the importance of sensor placement, a completely contrasting situation was taken into account. Both sensors were placed close to the free end and the system was analysed. Fig. 26 shows the shape estimation done with two sensors placed close to the free end(red dots indicate sensors), the theoretical model, the positional error(in x and y directions) along the length of the beam and the experimental setup of the beam bending due to a load of 0.48N. The shape estimation is not as erroneous as in the previous case(fig. 25). The error is in the magnitude of tens of centimeters. The shape of the beam has been well-estimated, given that the sensors are not yet optimally placed.

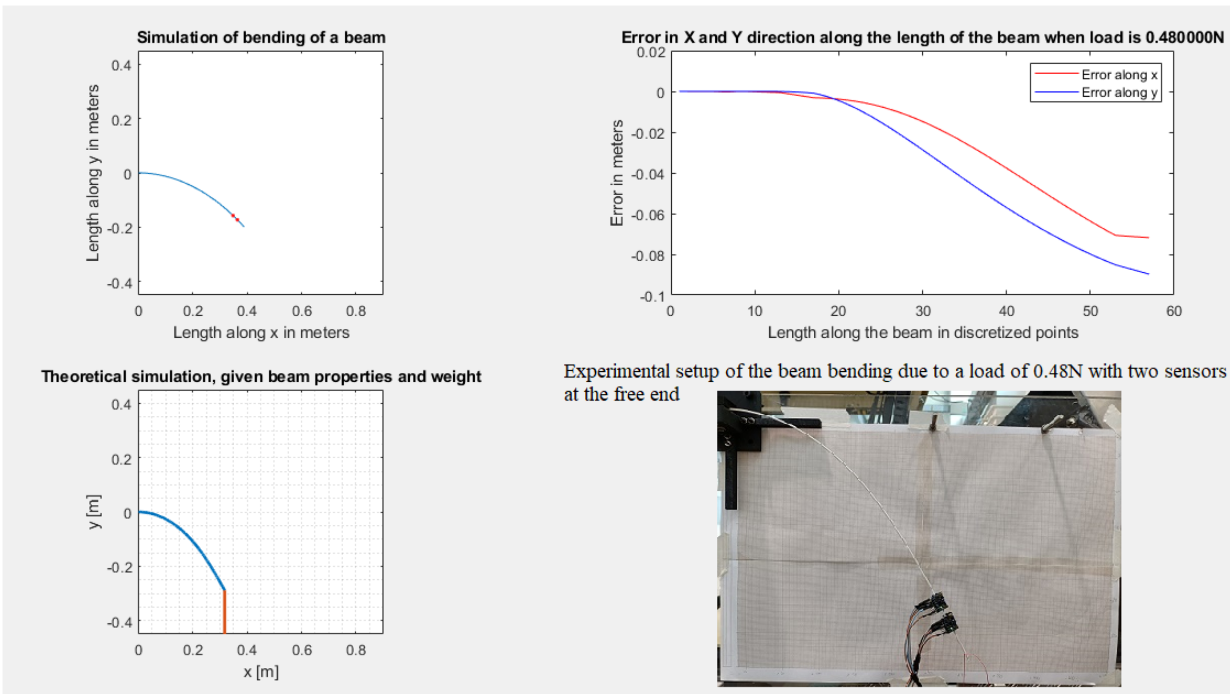


Figure 26: The shape estimation done with two sensors placed close to the free end(red dots indicate sensors)(top left), the theoretical model(bottom left), the positional error(in x and y directions) along the length of the beam(top right) and the experimental setup of the beam bending due to a load of 0.48N(bottom right)

Now we consider another case, where we find the optimal sensor placements given the length of the beam and the weight at the free end, then place the sensors at these optimal locations and subsequently estimate the shape of the beam in real-time and also see how the error varies. Fig. 27 and fig. 28 give the result of optimizing the sensor placement along the length of the beam and also shows how the error varies as the placements of the sensors vary. Fig. 29 shows the shape estimation of the beam, the positional error(in x and y directions) along the length of the beam, the theoretical model and the experimental setup when the beam is loaded with 49gms at the free end. As we can see in fig. 29, the positional error along the length of the beam is only in the magnitude of five to fifteen millimeters compared to the error of tens of centimeters in the previous case (fig. 26). Hence, the error has significantly reduced when the sensors placements have been optimized.

```
lengthOfBeam =

    0.4500


load =

    0.4900


bestSensorOnePos =

    0.1091


bestSensorTwoPos =

    0.3091


bestError =

   4.1382e-10
```

Figure 27: Print statement of optimum sensor placement when beam length is 45cms and load is 0.49N

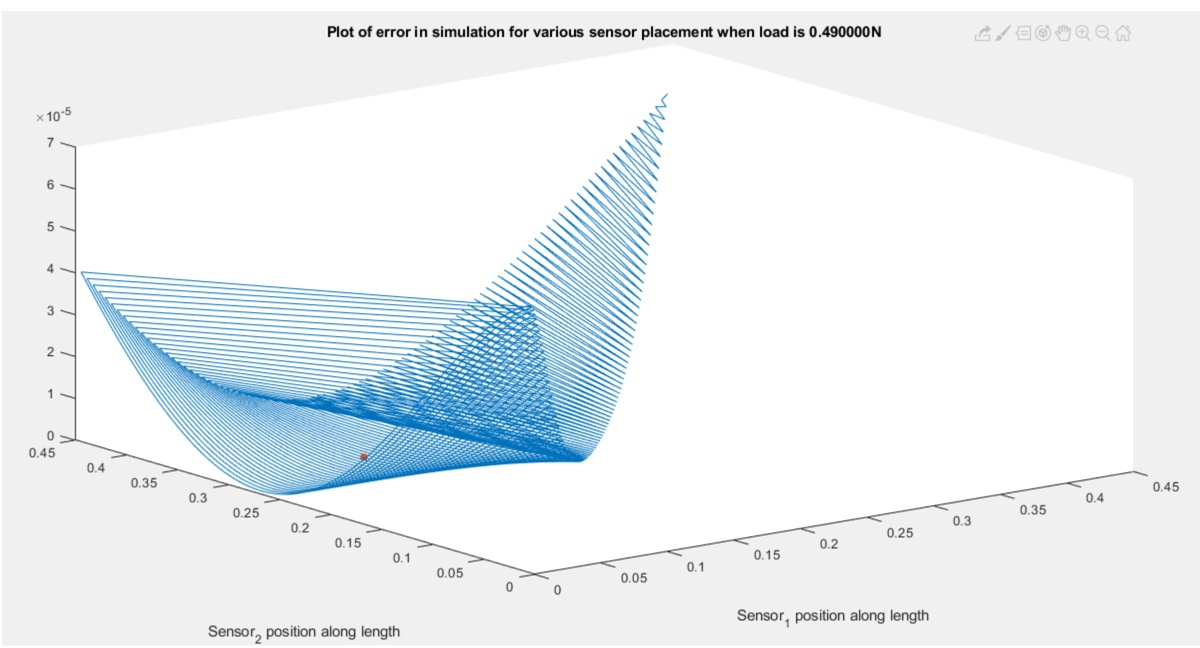


Figure 28: Plot of variation of shape estimation error against sensor placement with optimum sensor placement marked(red dot)

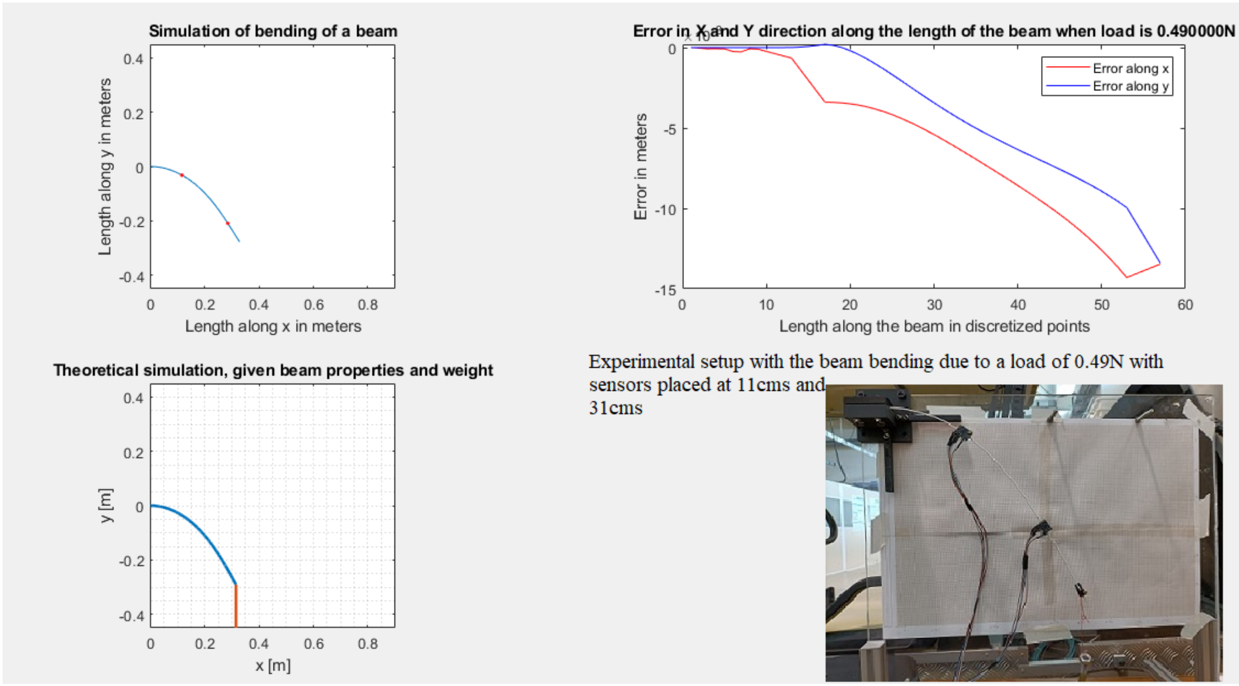


Figure 29: The shape estimation done with two sensors placed close to the free end(red dots indicate sensors)(top left), the theoretical model(bottom left), the positional error(in x and y directions) along the length of the beam(top right) and the experimental setup of the beam bending due to a load of 0.49N(bottom right)

Also, during the course of this project, it was verified that as the loading conditions of the beam changes(as the load at the free end changes), the optimum sensor locations changes. Fig. 30 shows how the optimum sensor locations varies with the load at the free end of the beam.

During the course of this project, it was also observed that the shape estimation error reduces as the loading at the free end increases. This can be attributed to the fact that when the load is low, the weight of the sensors and the wire(which is very low) has a larger influence on the shape estimation. But as the load increases more and more the influence of this weight becomes negligable.

These results show that with the optimal sensor placement, which has been done during the course of this project, IMUs prove to be a very good choice as sensors for the particular application of shape estimation of bending beams in continuum robots. Even though they may have some disadvantages like the requirement of long wires, drift of accuracy and response as time passes, they have many advantages as in they produce real-time(for some time), repeatable, accurate results, and they do not influence the bending of the beams.

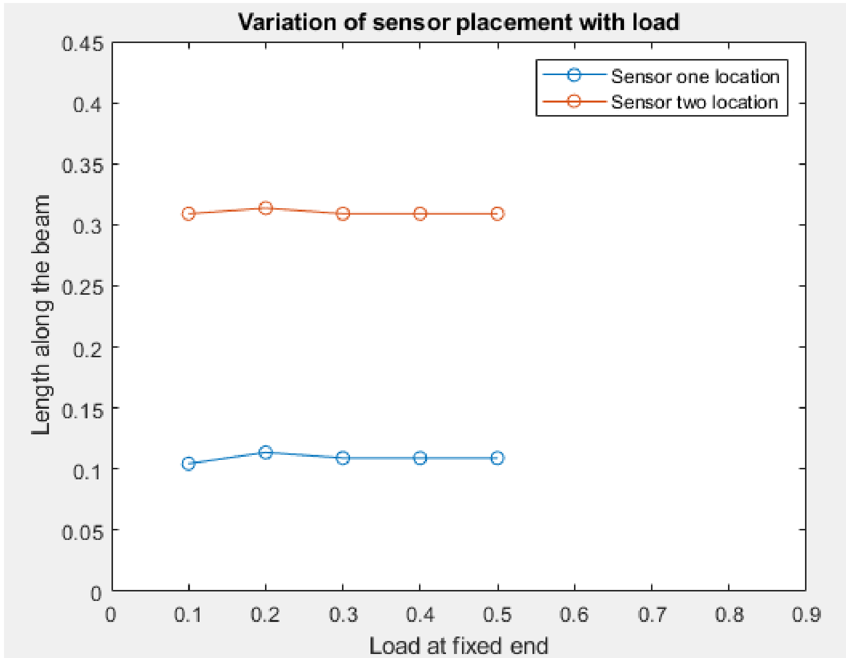


Figure 30: Variation of optimum sensor location(in meters)with load

Hence, IMUs may be a good choice of sensors, instead of resistive force sensors or flex sensors, for continuum robots using bendable beams as their links as they provide approximate real-time models of the beams. These models may be used to create a feed-back loop to correctly estimate and control the position of the end-effector controlled by the beams. There may definitely be many more applications where these methods may find use.

6 Conclusion

After an extensive literature survey, we have understood and appreciated the state of the art in the field of continuum robots and subsequently realised the importance of state estimation of the robot links/arms in improving the control & accuracy of the system. This leads us to the sensor based state estimation of a continuum robot which is what we explored during the course of this project.

Two types of sensing methods have been tested - a resistive force sensor and an inertial measurement unit (IMU). The use of IMUs has been found to give the best results.

We have developed a theoretical model to give us the state of a cantilever beam with a point load at the free end which is usually the case encountered in a beam based parallel continuum robot. Subsequently, we have developed a second experimental model to take rotation gyro values from the IMUs fixed at specific location on the beam to solve for the final state (position) of the beam. All of the above models are strictly planar i.e. a 2D plane.

For a given load, the results of the two models are compared to give an estimate of the error. Our models have been used and tested for a 2 sensor system as we practically implemented only for 2 sensors. Nonetheless, the models can be used for systems with a higher number of sensors.

Next, an attempt at solving the problem of optimizing the sensor placement to minimize the error between the two models has been made which has yielded good results. We have analysed our results and understood that all our models including the optimisation are satisfactory and can be visually validated. This doesn't eliminate the fact that small errors inadvertently creep in as there are certain unaccounted loads and disturbances in reality such as the weight of the sensors & cables, point load not being exactly in the end point, certain assumptions taken in the model, etc. which affect the final result.

This project helped us explore and understand the growing field of continuum robotics and its wide range of applications and also the challenges currently faced by the research community. We believe that our work will further the cause of robotics and we aspire to develop smarter solutions to challenges faced by humanity through the means of technology & robots.

7 Future Scope

The future scope for this work can be seen below:

- The number of sensors used can be increased to more than 2 for better accuracy of the estimation. In this project, we used only 2 sensors to demonstrate the proof of concept.
- The optimisation for sensor placement should be extended to cases with more than two sensors. The present program can only solve for 2 sensors.
- Shape estimation for spatial beams is an exciting avenue to further this work. Only planar 2D case is solved in this project.
- The real interesting step next for us would be to install this in a planar parallel continuum robot and develop a feedback control system for the robot. This possibility excites us as we would be able to see a practical working application for our work.

Bibliography

[1] P Andorf et al. "Robot finger". In: *URL: http://www. freepatentsonline. com/3981528. html, RL(1971). Gripping device. URL: http://www. freepatentsonline. com/3601442. html* (1976).

[2] ArduinoGetStarted. `https://arduinogetstarted.com/tutorials/arduino-force-sensor`.

[3] Jessica Burgner-Kahrs, D Caleb Rucker, and Howie Choset. "Continuum robots for medical applications: A survey". In: *IEEE Transactions on Robotics* 31.6 (2015), pp. 1261–1280.

[4] Hao Cheng et al. "Orientation to Pose: Continuum Robots Shape Sensing Based on Piecewise Polynomial Curvature Model". In: *arXiv preprint arXiv:2103.05150* (2021).

[5] Components101. `https://components101.com/sensors/mpu6050-module`.

[6] JBC Davies. "Dextrous manipulator for complex objects". In: *Proceedings Fifth World Conference on Robotics Research*. 1994, pp. 17–28.

[7] Hunter B Gilbert, D Caleb Rucker, and Robert J Webster III. "Concentric tube robots: The state of the art and future directions". In: *Robotics Research* (2016), pp. 253–269.

[8] Shigeo Hirose. "Biologically inspired robots". In: *Snake-Like Locomotors and Manipulators* (1993).

[9] Phillip Hyatt et al. "Configuration estimation for accurate position control of large-scale soft robots". In: *IEEE/ASME Transactions on Mechatronics* 24.1 (2018), pp. 88–99.

[10] Instructables. `https://www.instructables.com/MPU-6050-Tutorial-How-to-Program-MPU-6050-With-Ard/`.

[11] Invensense. `https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/`.

[12] Mircea Ivanescu and Viorel Stoian. "A variable structure controller for a tentacle manipulator". In: *Proceedings of 1995 IEEE international conference on robotics and automation*. Vol. 3. IEEE. 1995, pp. 3155–3160.

[13] Beobkyoon Kim et al. "Optimizing curvature sensor placement for fast, accurate shape sensing of continuum robots". In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 5374–5379.

[14] Srikanth Kolachalama and Sridhar Lakshmanan. "Continuum robots for manipulation applications: a survey". In: *Journal of Robotics* 2020 (2020).

[15] Arthur W Mahoney et al. "On the inseparable nature of sensor selection, sensor placement, and state estimation for continuum robots or "where to put your sensors and how to use them"". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 4472–4478.

[16] Bo Ouyang, Yunhui Liu, and Dong Sun. "Design of a three-segment continuum robot for minimally invasive surgery". In: *Robotics and biomimetics* 3.1 (2016), pp. 1–4.

[17] Selim Ozel et al. “A composite soft bending actuation module with integrated curvature sensing”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 4963–4968.

[18] Graham Robinson and J Bruce C Davies. “Continuum robots-a state of the art”. In: *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*. Vol. 4. IEEE. 1999, pp. 2849–2854.

[19] Jeff Rowberg. `https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050`.

[20] Daniel Caleb Rucker. *The mechanics of continuum robots: model-based sensing and control*. Vanderbilt University, 2011.

[21] P-L Schaefer et al. “Strain gauges based 3d shape monitoring of beam structures using finite width gauge model”. In: *Experimental Techniques* 43.5 (2019), pp. 599–611.

[22] Chaoyang Shi et al. “Shape sensing techniques for continuum robots in minimally invasive surgery: A survey”. In: *IEEE Transactions on Biomedical Engineering* 64.8 (2016), pp. 1665–1678.

[23] Sparkfun. `https://www.sparkfun.com/products/10264`.

[24] Hongbo Wang, Massimo Totaro, and Lucia Beccai. “Toward perceptive soft robots: Progress and challenges”. In: *Advanced Science* 5.9 (2018), p. 1800541.

[25] Wikipedia. `https://en.wikipedia.org/wiki/Continuum_robot`.