NAZARBAYEV UNIVERSITY

School of Engineering and Digital Sciences

Bachelor of Engineering in

Mechanical and Aerospace Engineering

AI ENHANCED FLEXIBLE MODULAR SENSORS

(Final Report for Capstone Project)

by

Daulet Sagyngali
Nurzhan Nurguatov
Yerzhan Akzhigitov

Principal Supervisor: Gulnur Kalimuldina
Co-supervisor: Azamat Yeshmukhametov

April 2025

Declaration

We hereby declare that this report entitled “AI Enhanced Flexible Modular Sensors” is the result of our own project work except for quotations and citations that have been duly acknowledged. We also declare that it has not been previously or concurrently submitted for any other degree at Nazarbayev University.

Names: Daulet Sagyngali, Nurzhan Nurguatov, Yerzhan Akzhigitov

Date: April 29, 2025

Abstract

Recent robotics literature primarily focuses on continuum robots, which demonstrate high levels of versatility and flexibility compared to traditional rigid-link robots. Many widely used continuum robots are tendon-driven and actuated by motors. This project aims to design a continuum robot capable of reaching arbitrary 3D coordinates by developing and integrating strain sensors to identify the robot's position. This work introduces novelty through a less conventional continuum robot structure that is designed to retain key performance characteristics similar to standard approaches. The robot's backbone structure comprises four identical sections constructed using ball-and-socket joints. A flexible strain sensor is fabricated using Ecoflex and multi-walled carbon nanotubes (MWCNTs). Electromechanical characterization of the sensors demonstrated high linearity ($R^2 = 0.981$) up to 100% strain, a gauge factor of 4.12, and low hysteresis (1.59%). Successful integration onto the robot structure enabled the correlation of sensor resistance with robot bending. This study validates the feasibility of the proposed robot design and sensor configuration, providing a basis for the future implementation of machine learning algorithms for automated control.

Table of contents

1. Introduction

The opportunities of robotics have ignited many bright scholars and inventors to investigate the possibilities of implementing machines in various fields. Solely the manipulators were able to revolutionize the automation process in various sectors of industry, but the flaw of the manipulator was the rigid link structure which meant that the robots often lacked the versatility in positions apart from the floor of the manufacturing plants. From this point, an alternative to rigid-link structured robot manipulators was introduced, an invertebrate-like robot manipulator that was termed as "continuum robot", a concept that would be an excellent solution for many issues, as the continuum robot was able to bend, extend, contract and twist depending on the potential implementation requirements alongside the whole of the structure of the robot. Moreover the continuum robot has a classification based on the design: tendon-based, concentric tube, locally actuated backbone, and variable backbone stiffness [1], however the more widely implemented type of continuum robot utilizes a sole secure backbone that will act similarly to a bionic snake bone and have tendon cables as muscle tissues and these types of robots are extremely versatile and effective in unstructured conditions [2]. A schematic diagram of the tendon driven continuum robot is shown in **Figure 1** The reason for the success and wide spreading of the tendon-driven continuum robots lies in its wires: firstly the wire itself is highly flexible which allows it to come up with different designs for the continuum robot without suffering in performance in any given aspect. Secondly, the tendons are located along the whole structure which in return means that the transmitted force from the actuators is sufficient to control the robot from one point. Albeit the area where the tendon-based continuum lacks is the tension during the bending process and will most likely require additional tension control mechanisms to cope with the arising issue [3]. Another point to pay attention to is the fact that in most of the designs present the implemented backbone movement is actuated by bending the backbone at the disjointed positions, subsequently, this allows the continuum robot structure to have more than ample degrees of freedom overall, however locally at each joint typically the robot will lack the necessary degree of freedom in comparison to a biological alternative [4].

To enhance the perception and control capabilities of tendon-driven continuum robots, the integration of sensing technologies is crucial. Wearable strain sensors, in particular,

offer a promising solution for monitoring the robot's deformation and internal forces. The working principle of wearable strain sensors is based on the conversion of mechanical deformation into an electrical signal. There are several types of strain sensors which include resistive [9], capacitive [10], triboelectric [12], and piezoelectric [11]. The interest and need for those types of sensors are only growing yearly because of their mechanical flexibility, high sensitivity, and durability. Recent studies show that the flexible sensors market will increase from $3.6 billion to $7.6 billion by 2027 [13]. This indicates that wearable sensors are not only the future but already the present, which inspires confidence. **Figure 2** shows the possible applications of flexible sensors.

The main advantage of continuum robots is their versatility and flexibility. Consequently, the continuum robots excel in narrow and complex spaced environments. As a result, many scholars have been successfully manufacturing and developing the continuum robots to be implemented in the case of a disaster and search and rescue missions. A continuum robot was utilized when disaster struck Japan with the Kumamoto earthquake alongside the Fukushima Daiichi nuclear power plant catastrophe to scrutinize the collapsed house holdings [5]. The continuum robots' applications are not strictly for search and rescue operations, as according to Nahar, Yanik and Walker (2017), a continuum robot tendril was able to overcome experiments and tasks in environments stimulating International Space Station conditions successfully doing spacecraft maintenance, data collection, and planetary exploration [6].

1.1 Aim

The aim of this project will be to design a continuum robot capable of inspecting and reaching arbitrary 3D coordinates in space. In addition it is crucial to mention that the robot will be equipped with a strain sensor that will be obtaining certain data and based on the data the positioning of the robot on the 3D coordinate system will be assessed. Understanding the positioning of the continuum robot will allow to identify and further achieve the opportunity to inspect and reach the mentioned arbitrary coordinates, possibly basing the robot design similarly as depicted in **Figure 1.**

1.2 Objectives

Many objectives were set and many are to be achieved; however, the main four objectives are the following:

- Designing a tendon-driven continuum robot
- Developing and integrating strain sensors to identify the robot's position
- Implementing machine learning control

Member:	Daulet	Nurzhan	Yerzhan
Responsibility:	Experimental setup, fabrication of CDCR, circuit connection, sensor values resistance data testing.	Motor control, pulley design, kinematic analysis and the robot manipulation	Sensor design, manufacturing and testing, circuit connection and sensor values to robot movement correlation testing

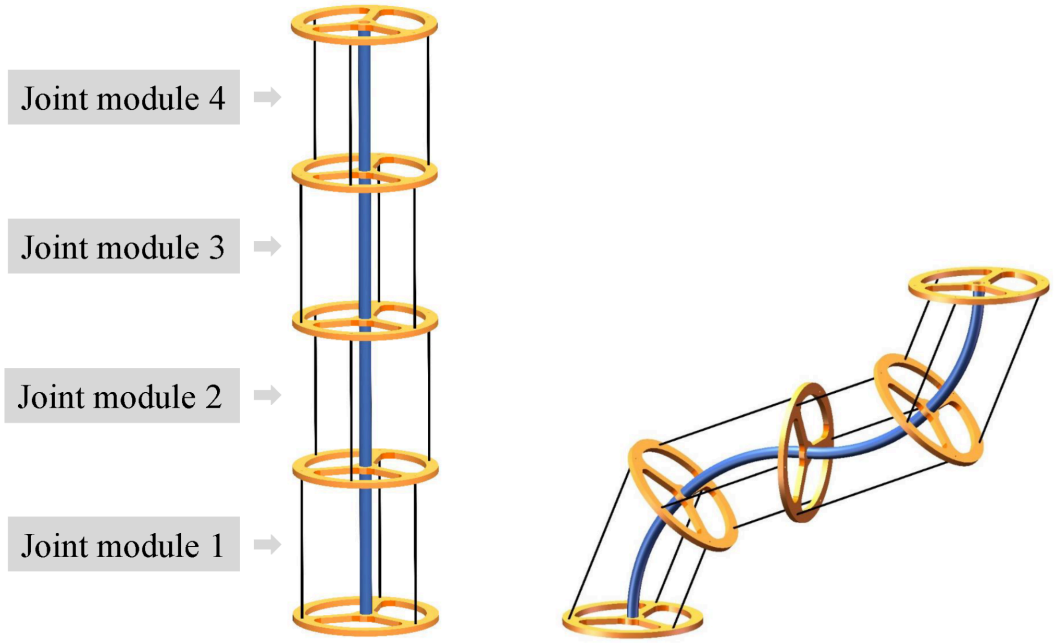

Figure 1. Schematic representation of the continuum robot: a) initial pose; b) bending position [7].

2. Literature Review

Cable-driven continuum robots (CDCRs) have been widely recognized for their ability to perform intricate and adaptable movements while maintaining compliance. Recent advancements in design, modeling, sensing, and control have focused on enhancing precision, responsiveness, and real-world applicability. Based on this characterization it was decided to investigate more advantages and disadvantages of the CDCR as a result to get a better grasp of the matter and to understand the complexity of which the CDCR offers during fabrication. Additionally, it is crucial to investigate the current state of the strain sensor market, as the strain sensor will be the key component of achieving the aim, thus it will be absolutely necessary to familiarize the current fabrication methods and further testing procedures that took place. researching on the matter will not only reinforce any ideas but also offer insightful information that will be crucial in testing the CDCR and allow it to offer a novelty in the current scientific field.

2.1 Robot Design

2.1.1 Backbone Structure

The backbone design of CDCRs is crucial in determining flexibility and structural strength, and achieving a balance between tensile and torsional stiffness, while bending flexibility remains a significant challenge. As a result, there are significant developments in the field as scholars consistently aim to improve the implemented designs of the CDCR's backbone, which typically will hold the construction on itself.

In the search to improve the backbones, the industry often outlines the importance of the backbone material to have high tensile and torsional stiffness, as such characteristics will minimize any complex deformations caused by the bending motion. Additionally some scholars have identified the importance of cable management in the tight and space constrained spaces which typically occur in the CDCR designs, in order to prevent any entanglement of cables, Qian et al. (2023) offers an intentional cable misalignment by 45° to prevent collisions, however this misalignment was done for CDCR designs where 4 cables are present on each joint and the case may change depending on the cable amount [14]. Moreover, the issue of coupling between the length of the cable and deformation of the joint was solved by implementing the Bowden tube [14] and this practical solution will be taken into consideration as the coupling issue will be a serious drawback during the design phase. Continuum robots

can operate without backbone, but flexible chambers are designed based on structures such as honeycomb [8] and origami [49]. According to Shen et al. (2024), there were two points to consider firstly is the implementation of thermoplastic polyurethane (TPU) material for printing and the backbone structure, as TPU has promising characteristics for being the backbone due to its low bending stiffness and high tensile and torsional stiffness, ensuring compliance without compromising structural integrity [7]. Albeit the more important point to consider was the implementation of the sensor, which was aligned on the backbone and was able to deliver data on the CDCR's positioning, which it was able to do so as being directly mounted on the backbone meant that any bending and deformation the sensor would immediately sense. Considering the designs mentioned above, the potential implementation of another unusual design was suggested, which brought to the ball-and-socket joint configuration made of ABS material as due to the high elasticity and rigidity the stiffness of the ball and socket joint allows the structure to remain straight even if external forces affect the CDCR [15]. However, a flaw of this design is the interference of the backbones and tendons, which requires calculations to ensure smooth movement without any interference [15].

2.1.2 Sensor Implementation

Accurate shape sensing is essential for effective motion control, particularly when external forces such as gravity and payloads are applied.

Stretchable capacitive sensors (RH-ESSA-01 from ElasTech) with a 50% stretch rate and 0.05% resolution have been used to achieve high sensitivity to bending and cost-efficient implementation, after, calibration procedure has been introduced to correct placement errors and demonstrate higher accuracy [7]. Elastic magnetoelectric strain sensors have been proposed as part of a kinematic model based on the product of exponential (POE) formula, demonstrating high precision and applicability in real-time scenarios, while motion capture experiments have validated their effectiveness [14].

2.1.3 Kinematic and Dynamic Modeling

Accurate kinematic and dynamic models are critical for predicting and controlling CDCR motions. The application of Cosserat rod theory has been shown to describe bending behavior effectively, with theoretical and experimental results achieving a match within a 2% error margin [16]. Optimization-based approaches have been

demonstrated to deliver faster computational results [16]. A dynamic model grounded in the principle of virtual power has been introduced, which accounts for cable constraints and friction effects, and the model has achieved real-time simulation efficiency with an average trajectory error of 1.92% [2]. A sliding-mode-based closed-loop control system has been implemented, reducing trajectory tracking errors by 82.94% compared to open-loop control [7]. An average stress strategy has been developed to address high-frequency dynamics, reducing computational stiffness and improving control accuracy [2].

2.2 Flexible Strain Sensor

2.2.1 Material Selection

Wearable strain sensors typically consist of a sensing part and a supporting part. The sensing part, often made of conductive materials like conductive polymers (PEDOT:PSS, PANI) or carbon-based nanomaterials (CNTs, graphene), undergoes changes in electrical resistance when subjected to mechanical strain. Conductive polymers offer good flexibility and biocompatibility, making them suitable for long-term wear. PEDOT:PSS(Poly(3,4-ethylenedioxythiophene):poly(styrenesulfonate), for instance, is a popular choice due to its high conductivity, environmental stability, and ease of processing [19]. PANI(Polyaniline), on the other hand, exhibits high sensitivity and mechanical strength, making it ideal for applications requiring precise strain measurements [20]. Carbon-based nanomaterials, such as CNTs(Carbon Nanotubes) and graphene, possess exceptional electrical conductivity and mechanical strength. CNTs, in particular, can be easily integrated into various substrates and exhibit excellent sensitivity, making them suitable for a wide range of applications [21]. Graphene, with its unique two-dimensional structure, offers high flexibility, transparency, and biocompatibility, making it ideal for wearable devices [22]. Metal nanowires, such as silver and gold nanowires, provide high conductivity and flexibility. However, they can be more challenging to fabricate and integrate into wearable devices compared to conductive polymers and carbon-based nanomaterials. Elastomers, such as silicone rubber, PDMS, polyurethane, Ecoflex, and TPU, offer high elasticity, durability, and biocompatibility, making them suitable for various applications. Silicone rubber is highly resistant to moisture and chemicals, making it ideal for long-term wear [23]. PDMS, known for its transparency and biocompatibility, is often used in

microfluidic devices and soft robotics [24]. Polyurethane is a versatile polymer that can be tailored to specific needs, offering a balance of strength, elasticity, and durability [25,26]. Ecoflex, a type of silicone-based elastomer, offers high elasticity and transparency, making it suitable for applications requiring high sensitivity [21, 27]. TPU, a thermoplastic elastomer, provides a good balance of elasticity, strength, and processability [28].

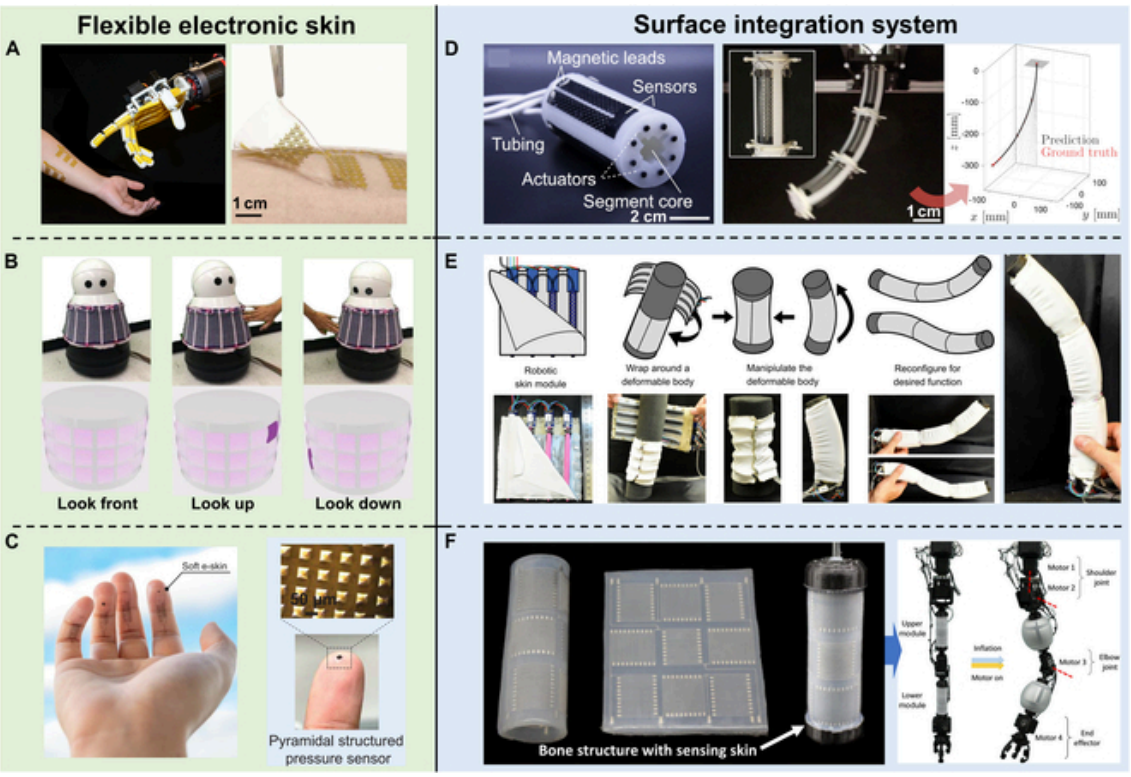

Figure 2. Various examples of flexible sensors and potential applications in soft robots [17].

2.2.2 Performance Parameters

In order to analyze the performance parameters of stretchable and flexible strain sensors, electromechanical characterizations are usually conducted using a variety of composite materials and structures. The common ones are sensibility, stretchability, linearity, hysteresis, stability, and durability.

2.2.2.1 Sensitivity and Stretchability

Sensitivity refers to the change in the output response of a sensor relative to a unit change in the input. For stretchable strain sensors, sensitivity is typically expressed through the gauge factor (GF), which is defined as the ratio of the relative change in the

sensor's resistance to the applied strain [18], [29], [30]. The GF for resistive strain sensors can be calculated using the formula:

$$GF = \frac{\Delta R/R_0}{\varepsilon} \quad (1)$$

In this equation, ε represents the applied strain, and R represents the resistance. A higher GF value indicates greater sensitivity to applied strain, which is desirable for detecting small strains within a linear operating range. Different composite materials and structural designs influence GF values. For instance, strain sensors based on carbon composite materials generally exhibit higher GF values compared to other composite classes. However, while some materials and designs can achieve extremely high GF values, such improvements often come at the expense of other critical performance factors such as stretchability, linearity, or transient response [31]. For example, the Au/PDMS composite strain sensor demonstrates a remarkably high GF of 5000, but its stretchability is limited to only 1% [32]. Certain advanced techniques, such as employing metal thin film crack engineering combined with conductive nano-reinforcements, enable the development of sensors with both high GF and reasonable performance trade-offs. For example, a braided graphene belt (BGG) strain sensor was reported to achieve an exceptionally high GF of 13,278 at a maximum strain of 55.55% [33].

Stretchability, another vital parameter, refers to the maximum strain a sensor can endure while maintaining a stable response. This characteristic varies significantly depending on the sensor's materials and structural interfaces. Some sensors exhibit very high stretchability, exceeding 900%, while others are limited depending on their design and composition [34].

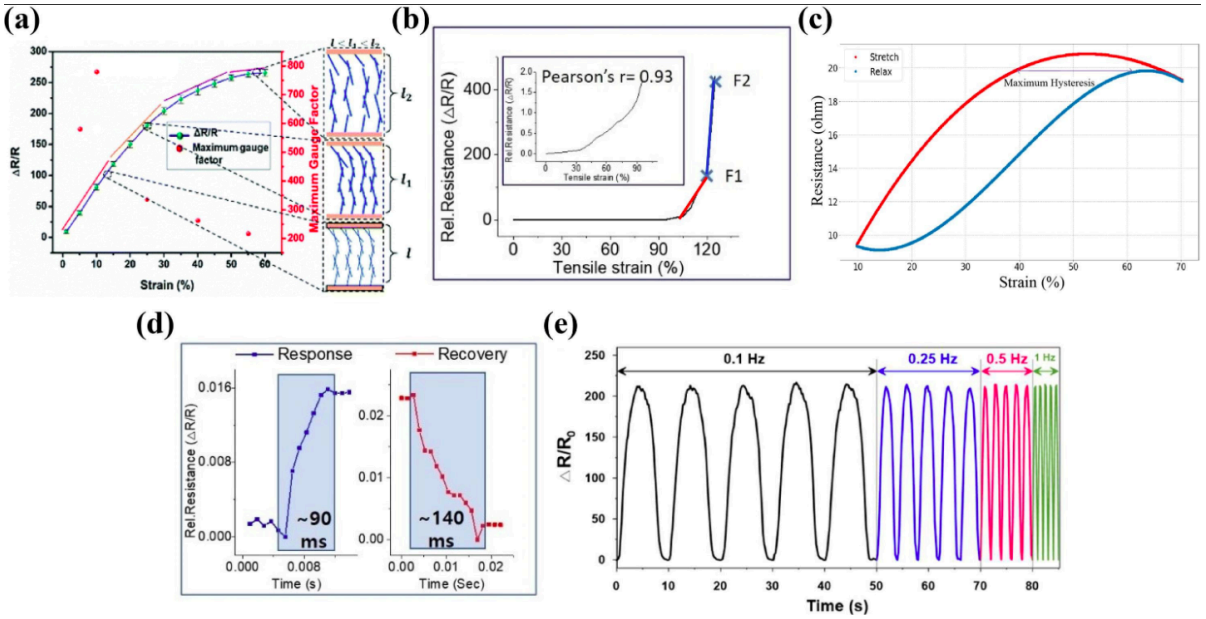

Figure 3. Electromechanical performance analysis of various stretchable nanocomposite-based strain sensors. (a) The aligned-CNT/PDMS composite strain sensor's stain response mechanism and gauge factor assessment at various strain values [31]. (b) Red and green lines represent the regions of e-skin LIG/PDMS/PSPI strain sensors where linear relative resistance changes. [35]. (c) Response of a conductive textile-based strain sensor test sample to hysteresis [36]. (d) Transient response of LIG/PDMS/PSPI strain sensor [35] measured by response and recovery times. (e) Stability analysis of Ag NPs/single-walled CNTs (SWCNTs)/polyester composite strain sensor at different loading frequencies of 0.1, 0.25, 0.5, and 1 Hz [37].

2.2.2.2 Linearity

A linear output response within the applied strain range is a crucial characteristic for strain sensors and is typically evaluated using linear regression modeling, quantified by the coefficient of determination (R^2). A higher R^2 value signifies a greater degree of linearity in the sensor's response, which is particularly desirable for precise measurements. Flexible strain sensors ideally exhibit a single, extensive linear region of operation combined with a broad sensing range or stretchability. However, non-homogeneous micro- or nano-deformations in the conductive networks or structural interfaces often result in non-linear behavior. This leads to multiple linear regions, each associated with a different gauge factor (GF) [38], [35], [33].

Many reported stretchable strain sensors display one, two, or three distinct linear regions in their output response. For example, Xie et al. developed a SWCNT-rGO/TPU

composite in a spiral yarn structure, which demonstrated five linear regions with a stretchability of 600% and a maximum GF of 2160 [38]. Similarly, a braided graphene belt strain sensor exhibited three linear regions with a 55.55% strain range and varying GFs across these regions [33]. **Figure 3b** illustrates an example of a linear relative resistance change in an e-skin LIG/PDMS/PSPI strain sensor. Two distinct linear regions are highlighted in red and green lines, corresponding to different operational ranges [35]. This visual representation underscores how the linearity analysis can reveal multiple response zones even within a single sensor. Although a single, ultra-wide linear range is considered ideal, it is not always a requirement for practical applications. Sensors with multiple linear regions can still effectively monitor human motion and other dynamic activities, offering sufficient performance for specific end-user needs.

2.2.2.3 Hysteresis

Hysteresis refers to the deviation in a sensor's response when the applied strain is removed, indicating a lag between the loading and unloading cycles. In resistive strain sensors, hysteresis is commonly observed due to the viscoelasticity of the composite material, unlike capacitive strain sensors, which generally do not face this issue. The degree of hysteresis is a critical parameter and is quantified as a percentage deviation using specific calculations, such as those applied to the Gly-KCl/Ecoflex composite strain sensor, where it was measured at 4.23% [39]. Sensors with significant hysteresis are unsuitable for wearable applications requiring consistent performance over numerous dynamic loading and unloading cycles. For instance, **Figure 3c** shows the hysteresis curve of a textile composite-based strain sensor as reported by Liang et al. [36]. This particular sensor exhibited a maximum hysteresis of around 20%, rendering it impractical for high-speed strain-sensing applications. Similarly, a butadiene-styrene rubber/natural rubber and graphene composite strain sensor fabricated by Lin et al. [40] demonstrated excellent properties, such as a GF of ~82.5 and 100% stretchability, but suffered from hysteresis with only a 75% elastic recovery. The hysteresis curves for this sensor were evaluated at various strain levels, including 30%, 60%, and 100%. Furthermore, according to research like those by Wang et al. [41], SWCNT/PDMS composite strain sensors that feature hierarchically reticulate SWCNTs implanted in PDMS have demonstrated hysteresis under cyclic stretching and relaxing conditions . For example, hysteresis was observed in MWCNT/PDMS nanocomposite strain sensors subjected to compressive loading and unloading. These behaviors highlight the

importance of minimizing hysteresis in strain sensors, especially for wearable devices. Ideally, hysteresis should either be negligible or quantifiable, allowing for its correction in the final output response to ensure reliable performance.

2.2.2.4 Stability and durability

In addition to other advantageous characteristics, maintaining a highly stable output response is crucial for stretchable strain sensors. The transient behavior described earlier can help evaluate the stability of such sensors, alongside cyclic loading and unloading tests. During these assessments, the sensor is subjected to a specific strain, and its response is monitored over an extended period to gauge output stability. However, resistive-type strain sensors often experience some instability due to overshoot phenomena and delays in reaching equilibrium when the load is applied or removed. For durability analysis, sensors undergo repeated stretching and relaxation cycles at fixed or variable strain rates. **Figure 3e** illustrates the cyclic loading and unloading performance of an ultrasensitive Ag NPs/SWCNTs/polyester woven strain sensor [37] tested at frequencies of 0.1, 0.25, 0.5, and 1 Hz. The MWCNT/PDMS sensor previously mentioned [41] regarding hysteresis also underwent 20 cycles at 30% strain with different MWCNT concentrations, demonstrating a sensitive and stable response compared to conventional high-threshold samples. Han et al. [42] developed a ternary hybrid composite strain sensor using nickel nanoparticles and graphene-coated PU sponges (Ni/GPUS), achieving a high gauge factor (GF) of 3360.09 and 60% stretchability. This sensor endured 1000 cycles at 30% strain, showing excellent durability, and its cyclic response was further examined at strains of 1%, 10%, 20%, and 35%. Similarly, Li et al. [43] introduced a graphene nanosheet (GNS), thin Au film, and GNS sandwich composite (GNSs/Au/GNSs) with a GF of 661.59. This sensor successfully completed 10,000 cycles at 50% strain for durability testing. Other composite stretchable strain sensors subjected to extensive cyclic durability tests [44], [45] have proven to be more reliable for wearable strain-sensing applications.

3. Methodology

After careful considerations from different scientific researches it was decided to develop simultaneously the CDCR and sensor and only after proper development and manufacturing of the parts to move on to the control.

Figure 4. Methodology of the project.

3.1 Continuum Robot Design

Using the solution offered by Zolfagharian et al., the potential design was decided to be a ball-and-socket joint design. As a result of this decision, the next part to decide was the manufacturing of the parts and due to its simplicity and overall effectiveness, the usage of the 3D printer from Ultimaker was pronounced. In addition to the printer decision, the material subsequently was taken into consideration and Polylactic acid (PLA) filament was the candidate alongside Thermoplastic polyurethane (TPU) filament; however, due to the abundance and convenience choosing PLA was the final decision. Afterwards, the crucial part of designing took place and using SolidWorks the design for both ball and socket joint parts was done. As the tendon driven continuum robot will require a hole for the cable, three distinct holes with a diameter of 3 mm and 120° apart from each other were made, the justifications for such decisions were that firstly, 120° perfectly suits the three holes and allows proper control for wires [47],

whilst the diameter was chosen 3 mm because to the fact of the metal cables being 2 mm and the property of the material which during printing and later after cooling had the tendency to shrink. Additionally, three more holes were added at the same 120° apart but slightly closer to the center of the disk for distinction but with a diameter of 3.5 mm for easier accommodation of M3x10 bolts for the proper alignment of spacer disks with each other. After obtaining every part and assembling them properly, the CDCR will require a platform on which the ball and socket joints will be placed and controlled via motors. Using the dimensions of the motors a CAD model will be made that will accommodate the motors and using M2.5 screws the motors will be secured and placed properly, after which ensuring the safe placement of the motor, the platform will be up for 3D printing with the PLA filament.

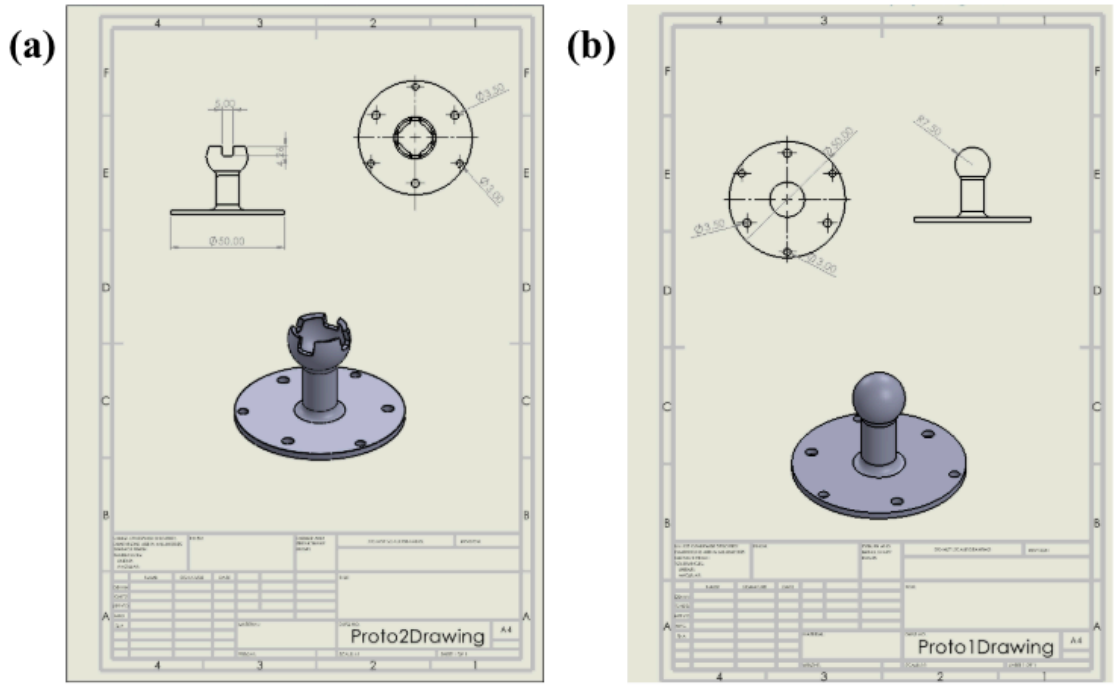

Figure 5. a) CAD model for socket joint; b) CAD model of ball joint.

3.2 Sensor preparation

3.2.1 Materials

To prepare samples, multi-walled carbon nanotubes (MWCNTs) with 95% purity, and an average diameter of 50-90 nm were sourced from Sigma-Aldrich Inc., USA. Hexadecyltrimethylammonium bromide (CTAB) with over 98% purity and molecular weight 364.46 g/mol from AMRESCO Inc., USA, was used to enhance functionality by surface modification. Ecoflex®00-30, purchased from Smooth-On, Inc., USA,

served as the polymer matrix, with a working time of 45 minutes and a specific gravity of 1.07 g/cc (ASTM D-1475). For filtering the modified MWCNTs, PTFE filter papers with a 0.22 µm pore size and 5.5 cm diameter were used. Carbon cloth was used as a conductor to connect conductive surfaces with wires.

3.2.2 Methods

3.2.2.1 Preparation of MWCNT and Ecoflex

The method for preparing the nanocomposite was adapted from the approach described by Aakyiir et al. [46]. Initially, 7 mg of CTAB dissolved in 10 ml of distilled water, followed by magnetic stirring for 5 minutes. After the CTAB was fully dissolved, 15mg of MWCNTs were added to the CTAB solution, and the mixture was sonicated in a water bath with some ice for one hour to prevent temperature increases due to ultrasonication. After sonication, the mixture was filtered using vacuum filtration and left to dry at room temperature for 10 hours. The resulting buckypaper was then cut into pieces measuring 30 mm in length and 10 mm in width, the maximum size allowed by the circular filter paper used.

For preparing the curing agent, the two components of Ecoflex were mixed in a 1:1 mass ratio and placed in a vacuum desiccator to remove any air bubbles. A total of 4 ml of Ecoflex was prepared by combining 2 ml from each component.

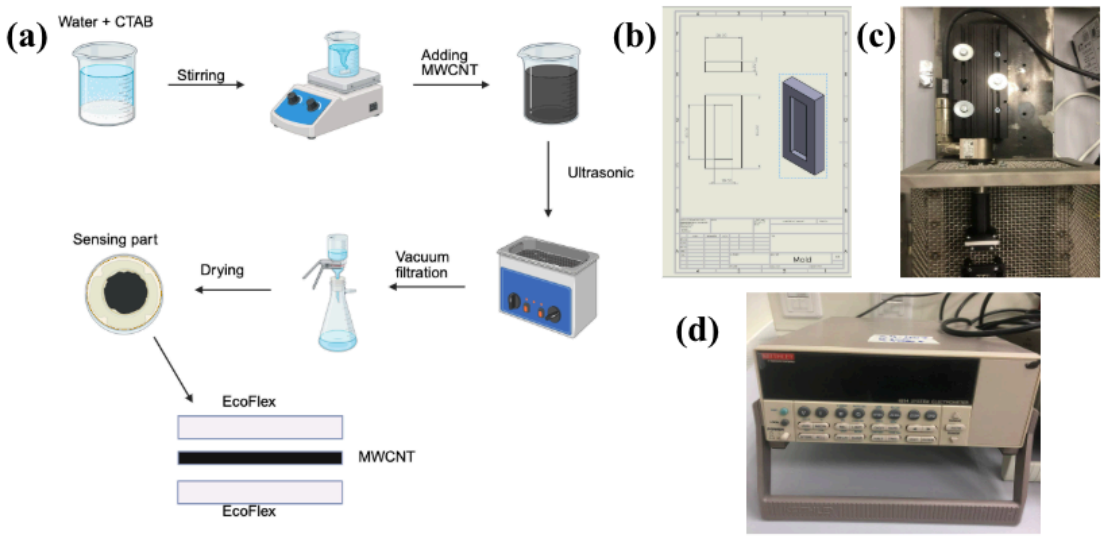

Figure 6. a) Schematic diagram of strain sensor preparation process; b) Mold CAD; c) Linmot motor; d) Keithley Kickstart.

3.2.2.2 Molding of the sensor

Bambu Lab 3D A1 mini printer was used to print the mold. The dimensions of the mold are in **Figure 6b**. The MWCNT paper was cut and placed into the mold. Then, 750 ml of Ecoflex was cast on the MWCNT paper. The complete solidification process of Ecoflex took 2 hours. After the curing process, filter paper was removed from the MWCNT, and carbon cloth was added to the sensor by casting the ecoflex on the second part of the conducting MWCNT film. So, the process was repeated for the opposite side.

3.2.3 Sensor Parameters Testing Setup

The testing setup for the sensor involves a Linmot motor, a Keithley 6514 programmable electrometer, and Keithley Kickstart Software, which collectively enable comprehensive performance evaluation. **Figure 6c** and **Figure 6d** illustrate the Linmot motor and Keithley electrometer, providing a visual representation of their integration into the testing system.

The Linmot motor, a linear actuator, operates by generating precise electromagnetic-driven linear motion. It applies controlled, periodic stretching and releasing of the sensor, simulating real-world mechanical deformation cycles. This setup allows for the assessment of the sensor's mechanical properties, including durability, stretchability, and hysteresis, by repeatedly extending the sensor to a predetermined length and returning it to its original position.

Simultaneously, the Keithley 6514 programmable electrometer monitors the sensor's electrical performance. This high-precision instrument measures resistance by applying a constant current and recording the corresponding voltage, enabling real-time tracking of resistance changes during deformation. These measurements provide valuable insights into the sensor's sensitivity, linearity, and stability under varying strain levels.

The testing process is streamlined by Keithley Kickstart Software, which automates data acquisition and analysis. The software configures measurement parameters, logs resistance data in real time, and stores results for further evaluation. Together, the Linmot motor, electrometer, and software form an integrated testing system, capable of thoroughly analyzing the functionality and reliability of the flexible strain sensor under dynamic conditions.

3.2.4 Sensor Fabrication

The results of the fabrication process are presented in this section, with accompanying images illustrating key steps. The chosen concentrations of 15 mg MWCNT and 7 mg CTAB were used consistently to ensure uniform sensor fabrication.

The process began with the preparation of a MWCNT sample as shown in **Figure 7a**, which was cut into the desired shape. The degasification of the Ecoflex matrix, depicted in **Figure 7b**, was carried out to eliminate air bubbles, ensuring a smooth and uniform surface. Following degasification, the first layer of Ecoflex was poured into the mold containing the MWCNT buckypaper. After curing, the filter paper was peeled off to reveal the integrated structure of the MWCNT and Ecoflex.

]

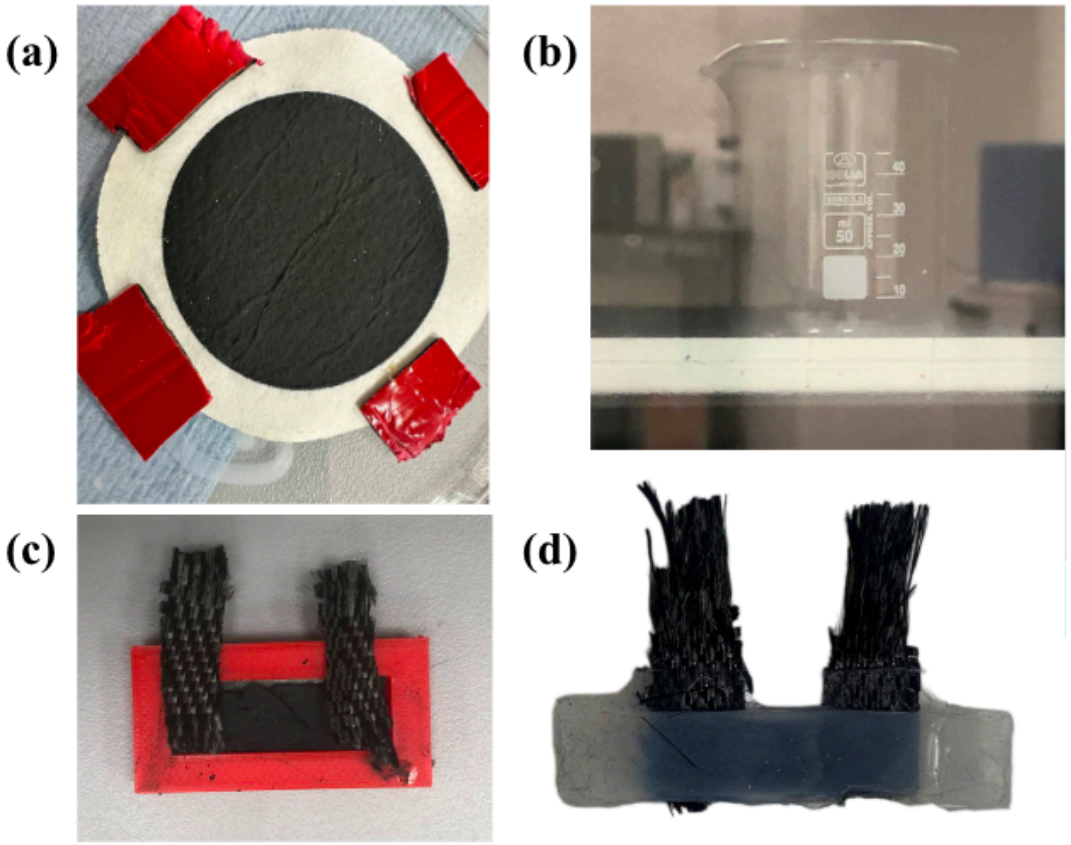

Figure 7. a) MWCNT Buckypaper; b) Bubbles Removing Process of Ecoflex; c) Carbon Cloth on Conducting Part; d) Flexible Strain Sensor

Once the first layer of Ecoflex was cured, carbon cloth wiring was directly attached to the MWCNT surface, ensuring a secure and conductive connection, as depicted in

Figure 7c. This approach provided a reliable electrical pathway while preserving the structural integrity of the sensor.

The nanocomposite underwent testing to assess its potential as a flexible strain sensor. Key performance parameters identified in the literature review were evaluated, and the results are detailed in the subsequent sections.

3.3 Motor Control

3.3.1 Motor Connection

The selected servo motors, Dynamixel MX-28, offer various control modes, including position and velocity control. Each motor is uniquely identified by an ID, allowing independent control of multiple motors. Using the software Dynamixel Wizard 2.0, three selected motors were initially scanned, and their IDs were configured to ID1, ID2, and ID3. To control these motors, parameters such as baud rate and protocol type were set up at a 57600 baud rate and protocol 1.0, respectively. The motors operate with a goal position range of 0 to 4095, where each unit corresponds to 0.088 degrees, enabling a full revolution at a goal position of 4095.

The Dynamixel motors for the tendon-driven continuum manipulator were controlled via packages based on *ros2_control*, and signals from the user interface were sent to the motors using the serial converter U2D2 [8]. Similarly, this project implemented the U2D2 converter and the DynamixelSDK library package, which contains all dynamixel motors' operating data, was used to approach the motors by plugging their ID numbers, as well as baud rate and protocol type. The library included "*read_write"* Python code, which reads the motor's position until it reaches the prescribed goal positions. The code had been designed for a single motor. Therefore, it was modified to control all three motors separately by identifying their ID number. Dynamixel MX-28 operates under 10 - 14.8 V, therefore, the selected motors were supplied with power at the recommended 12 V. **Figure 8** illustrates the hardware configuration of the motor actuation system.

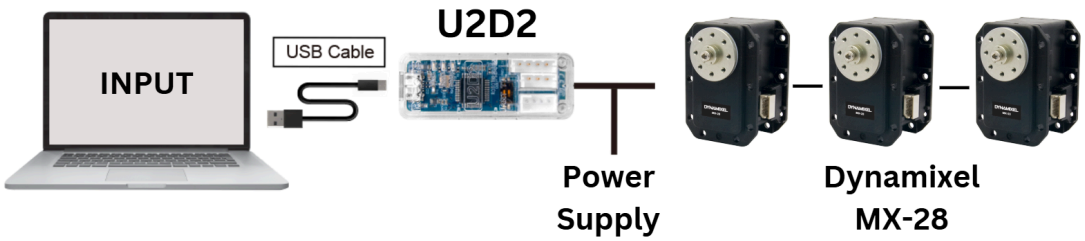

Figure 8. Motor actuation setup

3.3.2 Pulley Design

The design of the actuation pulley was determined by certain factors (**Figure 9**), resulting in a diameter of 28 mm. Primarily, vertical tendon alignment was required to prevent friction with guide hole edges. Additionally, constraints included the physical envelope of the robot and the limited position range of the motors. The pulley design includes 32 mm diameter rims intended to prevent the tendons from slipping off during operation. Holes on the pulley's cylindrical surface are designed for tendon attachment, while holes through the base surface facilitate mounting to the motor via bolts.

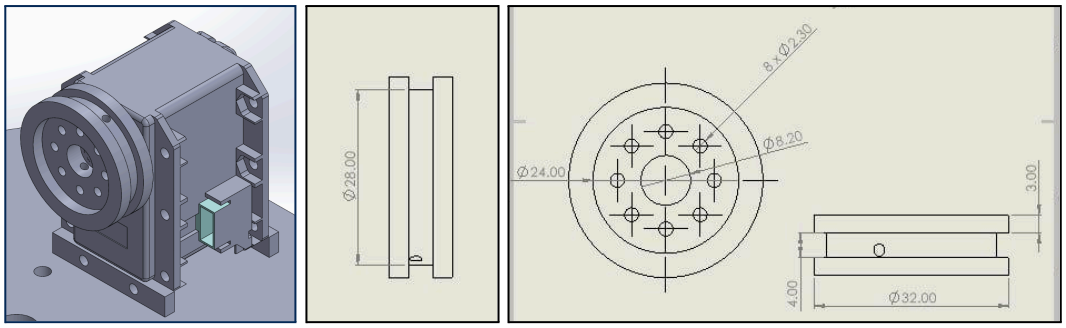

Figure 9. The 3D model and drawing of the tendon pulley attached to the motor

3.3.3 Kinematic Analysis

An optimal operational condition for the robot involves uniform bending across its sections. This uniformity dictates that all sections maintain identical bending directions and magnitudes throughout any flexural movement. Consequently, the modeling task reduces from analyzing the complete robotic structure to evaluating a single representative section. The robot's spatial configuration is defined by two primary angular parameters: the angle of rotation (observable from a top-down perspective as in **Figure 10b** and the angle of bending (viewed laterally) demonstrated in **Figure 10a**. These two parameters form the basis for determining the tendon displacements needed to attain the desired spatial configuration.

Figures 10 (c) and **(d)** illustrate a generalized bending configuration characterized by a specific angle of bending and angle of rotation. **Figure 10c** presents a lateral view of the first robot section, oriented such that the view's horizontal axis aligns with the direction of bending. Within this reference, the positions of key section components are defined using Cartesian coordinates, with the origin presumed to be at the center of the section's base.

Specifically, the tendon routing holes at the base have y-coordinates equal to zero in this view. Their x-coordinates, representing the projected positions onto the bending direction axis within this plane, are designated X1, X2, and X3. The coordinates of the ball joint center relative to the base center remain constant at (0, h) within this lateral view frame, irrespective of the bending angle's magnitude.

The top view (right picture) highlights the angle between the origin and bending direction which is already defined as the robot's angle of rotation (α) Subsequently, the specific values of the projected x-coordinates (X1, X2, X3) in the lateral view are determined by the angle of rotation (α) and the constant radial distance (r) of each tendon hole from the center of the base (eq. 1):

$$X1 = r \cdot cos(\alpha) \tag{2}$$

$$X2 = r \cdot cos(120 + \alpha) \tag{3}$$

$$X3 = r \cdot cos(240 + \alpha) \tag{4}$$

As depicted in **Figure 10c**, three tendon wires route through guide holes corresponding to the projected base positions X1, X2, and X3. Their respective lengths within the first robot section are designated t1, t2, and t3. Defining (Ω) as half the angle between adjacent spacer disk planes, the individual tendon lengths (t1, t2, t3) can be calculated using (eq. 2):

$$t_i = 2h \cdot cos(\Omega) - 2X_i \cdot sin(\Omega) \tag{5}$$

Substituting X1, X2, X3 with the values from eq. 2-4, the following equations emerge:

$$t1 = 2h \cdot cos(\Omega) - 2r \cdot cos(\alpha) \cdot sin(\Omega) \tag{6}$$

$$t2 = 2h \cdot cos(\Omega) - 2r \cdot cos(120 + \alpha) \cdot sin(\Omega) \tag{7}$$

$$t3 = 2h \cdot cos(\Omega) - 2r \cdot cos(240 + \alpha) \cdot sin(\Omega) \tag{8}$$

When a section is in its neutral (straight) state, the lengths of the traversing tendons are t1 = t2 = t3 = 2h, where 2h represents the axial length of one section. The principle of section uniformity dictates that the deformation is identical across all four sections. Consequently, the total change in length for each tendon (d1, d2, d3) required to move the entire robot from the straight configuration to a bent one is four times the change in length experienced by that tendon within a single section. Thus, the total length change is calculated as:

$$d_i = 4 \cdot (t_i - 2h) \tag{9}$$

$$d1 = 4 \cdot (2h \cdot cos(\Omega) - 2r \cdot cos(\alpha) \cdot sin(\Omega) - 2h) \tag{10}$$

$$d2 = 4 \cdot (2h \cdot cos(\Omega) - 2r \cdot cos(120 + \alpha) \cdot sin(\Omega) - 2h) \tag{11}$$

$$d3 = 4 \cdot (2h \cdot cos(\Omega) - 2r \cdot cos(240 + \alpha) \cdot sin(\Omega) - 2h) \tag{12}$$

The motors are oriented in the robot's platform such that counterclockwise rotation, corresponding to an increase in the position value, retracts the associated tendons, while clockwise rotation releases them. Given that the motors operate within a position range limited to 0-4095 steps, their neutral or initial position is set to the midpoint (2048). This configuration allows for sufficient range for both tendon retraction and release. Furthermore, the calculated total changes in tendon length (d_i) are converted into the required changes in motor steps, and the goal position values for the motors are expressed as:

$$n_i = 2048 - d_i \cdot 4096/(R_p \cdot 2\pi) \tag{13}$$

where, R_p corresponds to the pulley radius, which is 14 mm.

The values of h and r from **Figure 10(c-d)** are measured 25 mm and 22.5 mm, respectively. All the constant parameters used in the previous equations are declared at the start of the operating Python code. Remaining values are the angle of rotation (α) along with Ω. Based on the uniformity of the robot's sections, we state the following equation:

$$4 \cdot (2\Omega) = 180 - 2\theta \tag{14}$$

Consequently, the value of Ω is defined by the angle of bending (θ). In summary, the target motor positions are computed entirely from the specified angle of rotation (α) and angle of bending (θ).

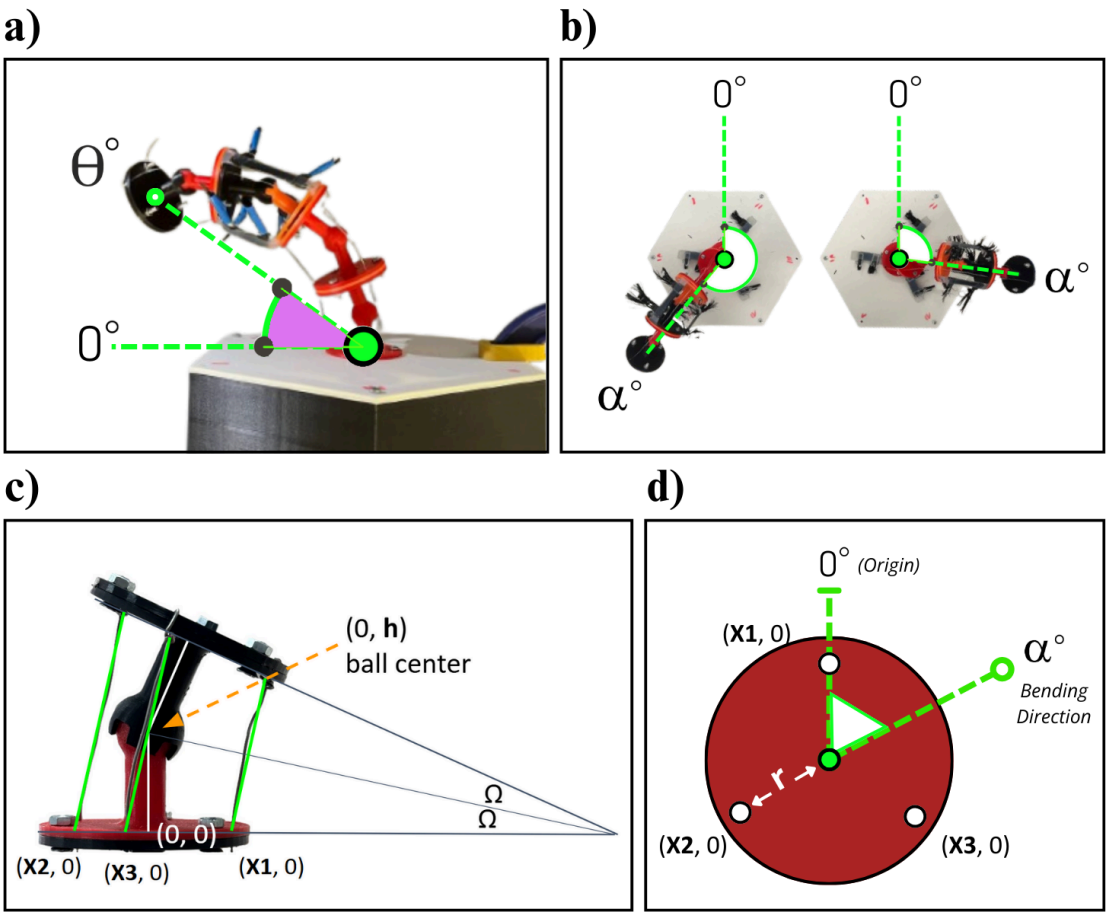

Figure 10. a) the robot's angle of bending (θ), viewed from the side; b) the robot's angle of rotation (α), viewed from the top; c) singular section positioning and corresponding parameters; d) the coordinate system and angles of each tendon wire in the CDCR base.

3.3.4 Joystick Connection

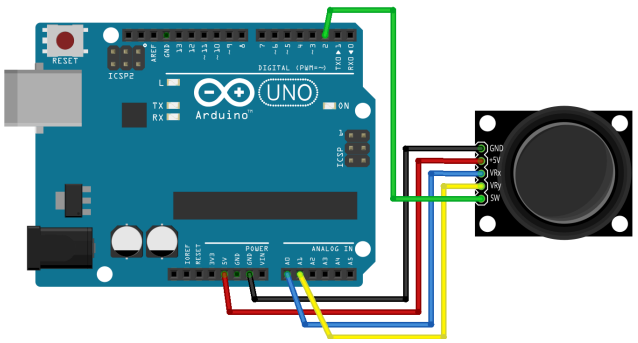

Figure 11. The joystick setup

Figure 11 illustrates the integration of an analog joystick module with an Arduino Uno microcontroller board, representing a common hardware setup for user input. The Arduino program was developed to acquire joystick positional data (X and Y axes). This program was subsequently enhanced to perform a Cartesian-to-polar coordinate transformation, yielding an angular position value which corresponds to the angle of rotation (α). Meanwhile, the value for the angle of bending (θ) is set manually. The existing read_write routine, responsible for motor actuation, was modified. Instead of commanding motors to predefined goal positions, this routine starts continuously receiving the calculated angular values from the Arduino program and dynamically sets the target positions for each motor based on this real-time input. Consequently, the positions of the motors are determined by angular inputs derived from the joystick's manipulation.

4. Results & Discussion

After thorough considerations of the current CDCR designs present on the marker it was decided that an infrequently used design would be suited. Overall the works of Zolfagharian et al. [15] were deemed interesting as the design of the ball and socket joint was not as widely spread as the typical CDCR design [15]. Albeit the flaws of the design offered by scholars were apparent, as the structure of the ball and socket joint was restricting the motion and potentially missing out the higher degree of freedom.

After detailed reviews of currently existing CDCR designs the ball and socket design was selected for this project. In comparison to the more traditional tendon driven or backbone employed designs of CDCR, the ball and socket design offers simplicity, modularity and flexibility as advantages, despite the rotational constraints and the subsequent limitation of degree of freedom as mentioned by Zolfagharian et al. [15]. The ball and socket design presents compromises between fabrication ease, motion range and structural integrity and these tradeoffs are acceptable considering the project objectives and time limits.

4.1 Continuum Robot Design

Utilizing the Ultimaker S5 3D printer, the following results were obtained, depicted in **Figure 12**. Printing the two parts would took approximately 2-3 hours and at the first the designs made faced the first coupling problem were the ball had a smaller diameter than the socket and as a result it would not stand straight, after some minor changes to the design, it was ready and able to be positioned straight without any external forces applied as shown in **Figure 12a**. However as to be able to withstand the given requirements the design of the socket was to be made slightly smaller ensuring the successfulness of the design, but another problem that have arised was the weakness of serrations of the socket which in result meant that during the assembly part of the design, those serrations would often break after enough forces were applied in order to wear the ball into the socket. The issue was partially resolved using a grinding machine which smoothed out the rough edges from the printing, after sorting out each part the next step of assembling the whole system came into place and as shown in **Figure 12b**, the system was able to withstand a straight up position without any interventions. Additionally it is worth mentioning that during the fabrication of parts a particular joint that will be placed on the top was fabricated slightly looser than the others as this

allows the CDCR more motion, whilst the joint on the bottom was made deliberately stiffer which allows the structure to be sturdy and not tilt during rest position. These manipulations were possible due to the use of the grinding machine. It showcased the range of motion of the system by bending. Overall this meant that generally the assembly of the CDCR was successful however, another crucial part was mounting the whole system, as the continuum robot was not able to withstand the forces it bore during bending and required assistance.

The ball and socket joint offers certain key advantages in comparison to the traditional continuum robot designs. Firstly, due to the nature of the ball and socket joint the fabrication of both components separately means the fabrication and assembly will be a much smoother process, in comparison to the traditional CDCR designs. The structure of the ball and socket joint does not require any support to maintain the given alignment, which will be necessary in case of a backbone based continuum robot as in some cases the continuum robot will require an additional actuator for maintaining the alignment, this would conflict with the project's objective of utilizing three actuator motors only.

Moreover, the use of the three-tendon system was favored due to increasing difficulty of motor control and kinematic analysis as the number of driving tendons increases [47]. Additionally it is worth mentioning that the implementation of the ball and socket joint design was dictated by the attempt to avoid overcomplicating the system by the need of a pneumatic actuator system.

Finally, the ball and socket joint system provides a higher degree of flexibility, offering better maneuverability in constrained and complex environments. Which is a serious advantage in comparison to the traditional backbone based design system which are typically constrained to specific planes [48]. Therefore, the ball and socket joint structure is a practical and efficient solution for achieving the set degrees of freedom considering the limitations and overall requirements of the project.

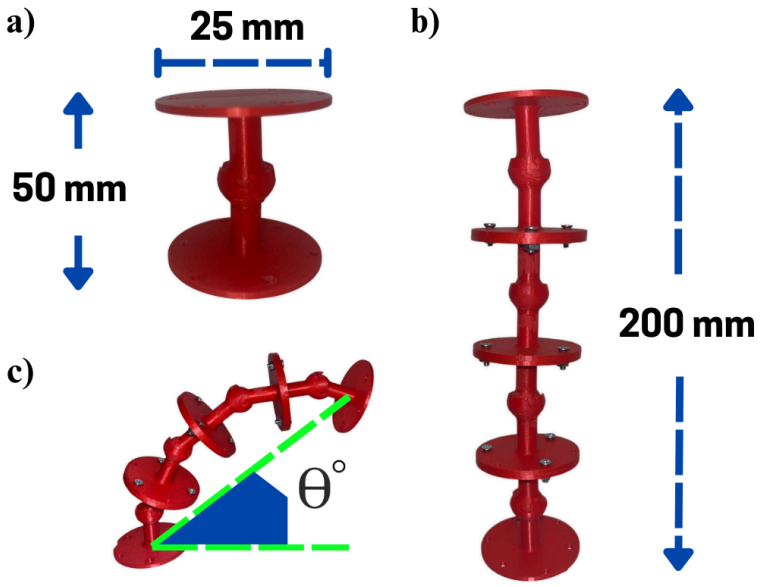

Figure 12. a) single backbone section; b) backbone with sections connected positioned straight; c) bending limit of the backbone structure

4.2 Sensor Performance Evaluation

In this section the performance characteristics of the developed sensors were assessed and analyzed in detail.

4.2.1 Linearity and Sensitivity

As seen in **Figure 13**, the sensor was subjected to 100% strain. Its linearity was equal to 0.981, meaning it has a high linearity between resistance and applied strain. Moreover, its gauge factor is equal to 4.12. 4.12 is a relatively high gauge factor, which means that the sensor has good sensitivity.

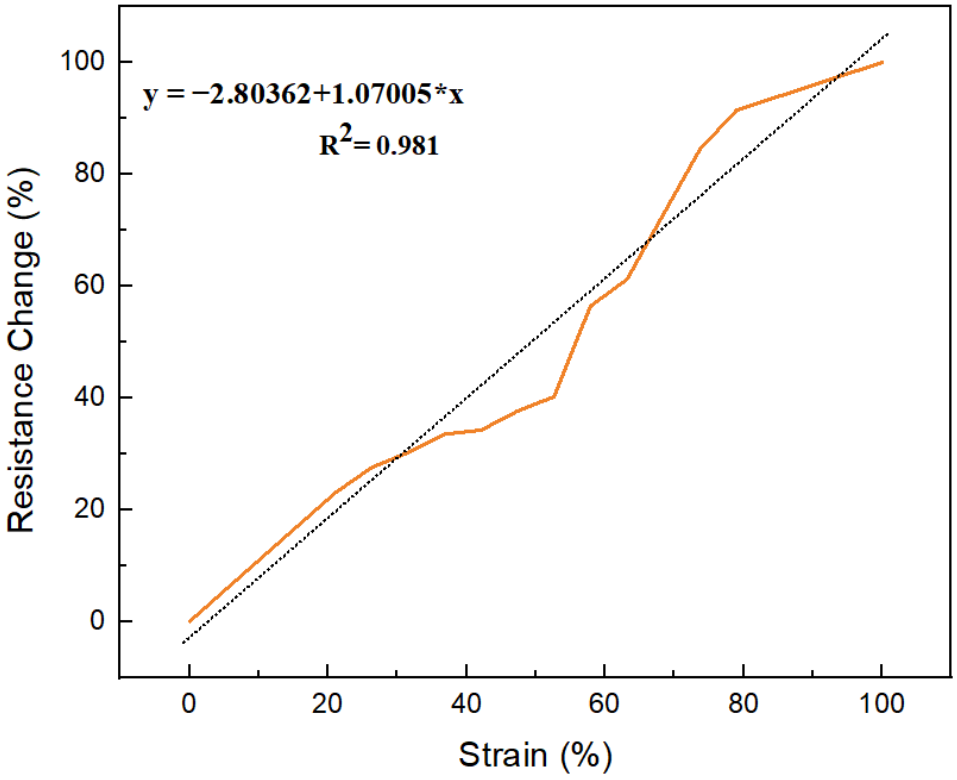

Figure 13. Strain response of MWCNT sensor

4.2.2 Hysteresis

Figure 16 illustrates the sensor's hysteresis behavior under loading and unloading cycles up to 100% strain. To determine the hysteresis value, the equation below was used:

$$H = \frac{Area_L - Area_{Un}}{Area_L} \times 100\% \quad (15)$$

The area under loading and unloading lines was calculated, and hysteresis was equal to 1.59%. A lower hysteresis loss is desirable as it suggests more accurate data after use of the sensor many times.

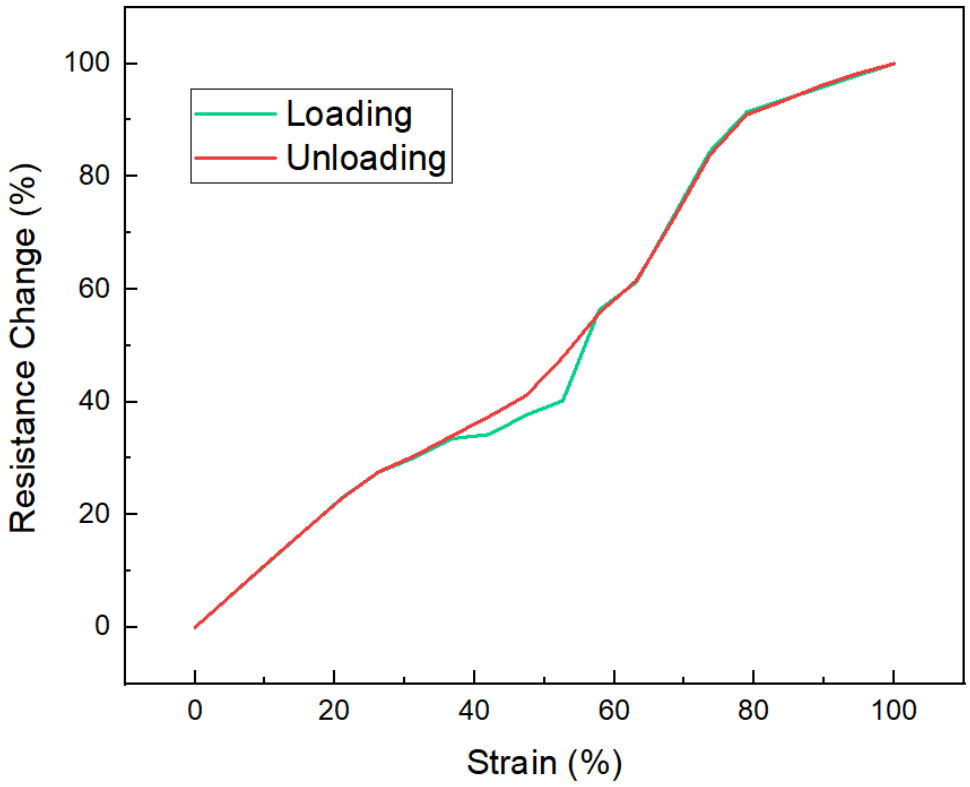

Figure 14. Hysteresis Response of Sensor

4.2.3 Stability

To determine the stability of the taken data, the sensor was stretched with the same amplitude of 100% strain and under the three different frequencies (1hz, 0.5hz, 0.25hz). From **Figure 15 (a-c)** it can be seen that the overall sensor was showing great stable resistance change, with small deviations in **Figure 15c**. Those deviations could be the result of some fabrication variations.

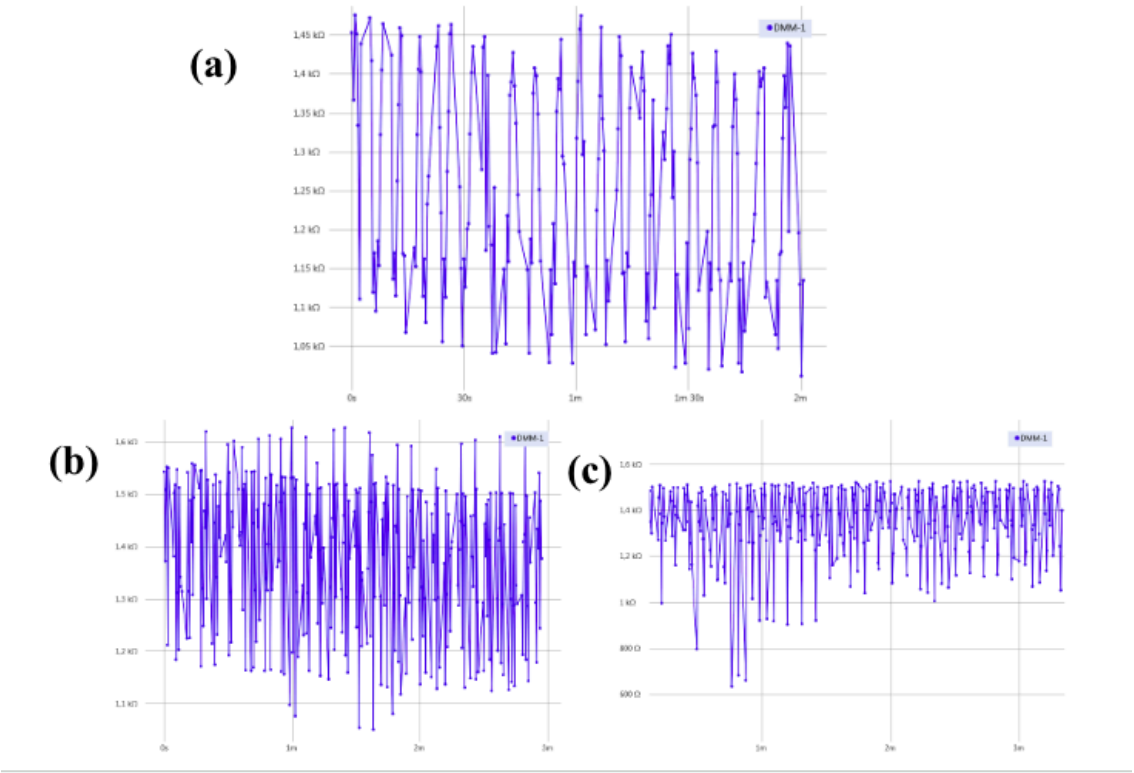

Figure 15. a) Resistance on 1 Hz; b) Resistance on 0.5 Hz; c) Resistance on 0.25 Hz at loading/unloading cycle.

4.3 Continuum Robot with the sensor incorporation

Additionally, it is important to note that during the testing and assembly stage of the continuum robot and sensor an issue arose: the fabricated first design of the strain sensor was insufficient in length, and the obtained data would be constantly incorrect and improper to interpret with positioning. As a solution, it was decided to alter the design of the sensor and a new design was offered, the mold of which is shown in the figure below:

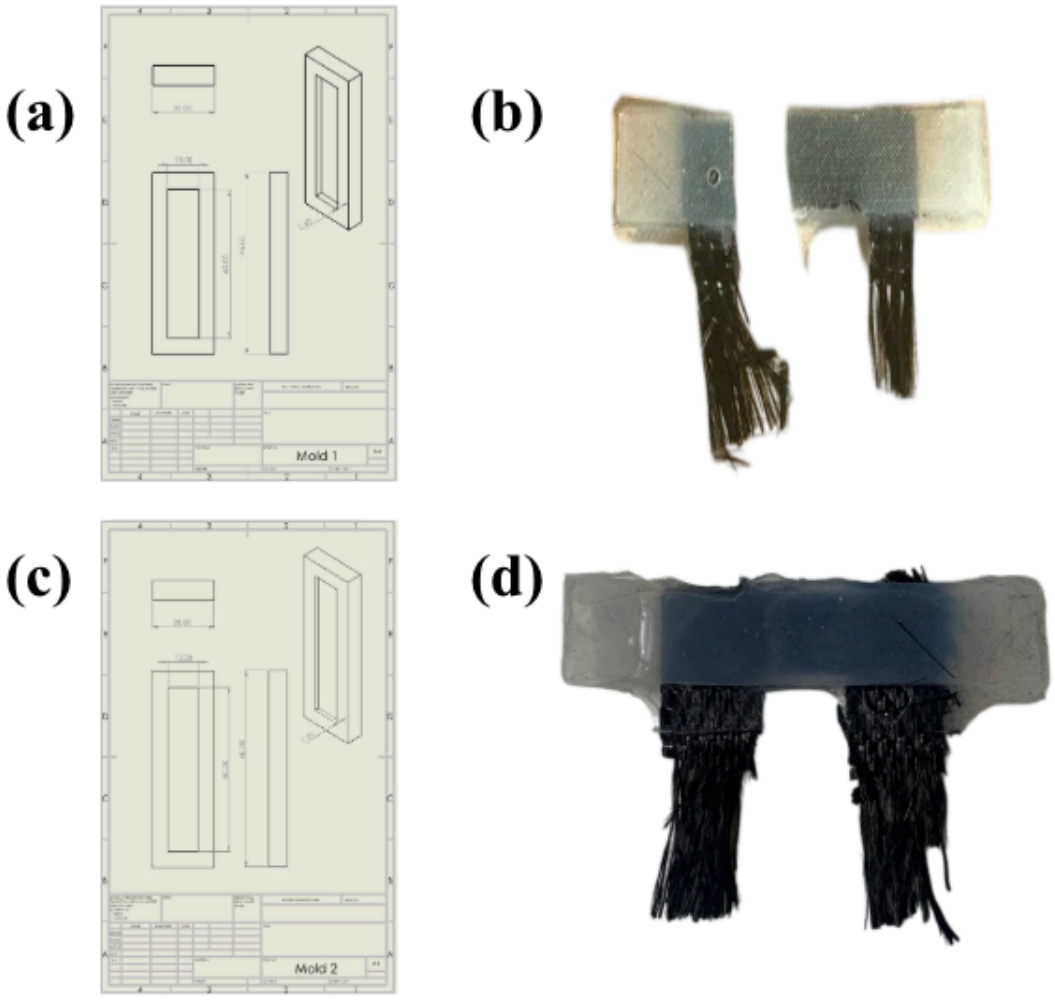

Figure 16. a) Mold of the second strain sensor prototype; b) second strain prototype; c) mold of the final strain sensor prototype; d) final strain sensor prototype.

As it is visible from the CAD drawing the dimensions of the fabricated strain sensor would be 45 mm x 10 mm x 5 mm, at first the dimensions were deemed acceptable however, further testings showed inferior results more specifically the strain sensor was not able to withhold the maximum strain during a sharp turn, as a result the strain sensor tore as shown in the **Figure 16a**.

Obtaining this result meant that the strain sensor design was to be altered once more and this time it was expedient to change the dimensions only slightly, as overall, the sensor exhibited good results. Considering this, the new design of the strain sensor offered is shown above:

Afterwards the fabrication the resulting strain sensor was ready to be tested on strain and resistivity data in the assembly with the sensor being of the following dimensions

50 mm x 10 mm x 3 mm. thinner sensors played not only the cosmetical role of making the sensors more eye pleasing, but also allowed to make the overall continuum robot construction more stable and less, additionally to obtaining more accurate data as now there is less ecoflex in the strain sensor thus less posible air bubbles that would interfere with the data collection, the fabricated new sensor is shown in **Figure 16d**.

4.3.1 Whole system assembly part

After obtaining the given results of manufacturing both the continuum robot and strain sensor, the next step was to assemble every component of the system together and test the robot's overall performance. According to Qian et al. [14] and Liu et al. [47], modular designs are much more beneficial rather than traditional non-modular, as the modular design of the CDCR will allow easier access and maintenance of any component of the robot, thus it was deemed a better way of fabrication. Consequently the lower base of the platform where the motors were mounted was able to easily withstand the torque or any kind of oscillation coming from the motor, the walls of the platform would be fabricated in such a way to offer the structural support for the main part and the upper component would be mounted on top of the walls and screwed for integrity as observed in **Figure 17**.

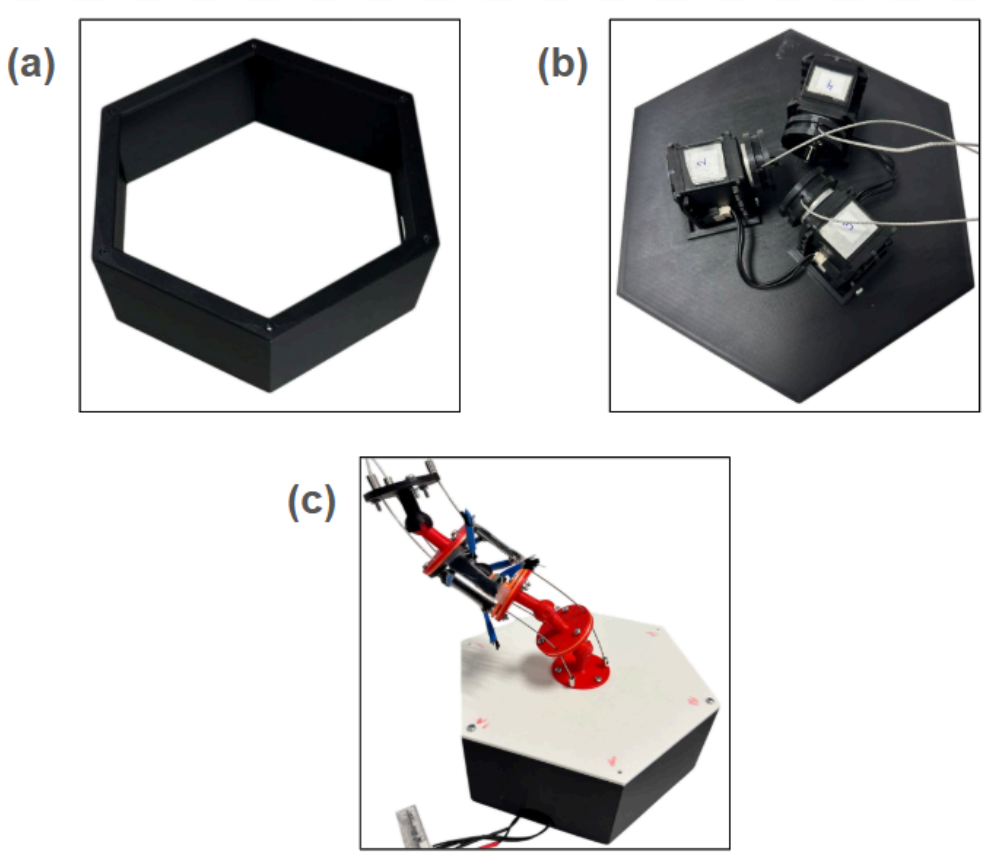

Figure 17. a) The walls of the platform; b) lower base of the platform with motors installed; c) Assembly of the whole platform.

At this stage, every part of the continuum robot was ready to be assembled and tested. During testing an issue arose that redefined the design of the continuum robot overall, as it was at this stage found that at the 3 section from the bottom the values of resistance was the highest alongside with the strain results, this meant that mounting the sensors on the third section was a better option as it would bring better results that are better to interpret. Empirically identifying the zone where most of the mechanical action will be taking place meant that the data obtained would be the most responsive and informative. As the project is a novelty, it was decided to focus on obtaining the range of data from one section allowing us to focus, analyze and correlate the data correctly, without the constant need to eliminate noise, and correlating the abundant resistance data from sensors from 4 sections. Additionally, it was identified that mounting three sensors on the same plane as where the ball and socket joint are located affected the values that are obtained, thus in between every section of the ball and socket joint a spacer disk was added so that the sensors would be mounted the same way without affecting both the motion of the robot and the resistance values it obtained. Controlling

the CDCR was done via connecting a metal wiring across the whole structure from the three motors The image of the fully assembled continuum robot is attached below:

4.3.2 Obtained Data

To receive real-time sensor data, an Arduino mega circuit was used as shown in **Figure 18**.

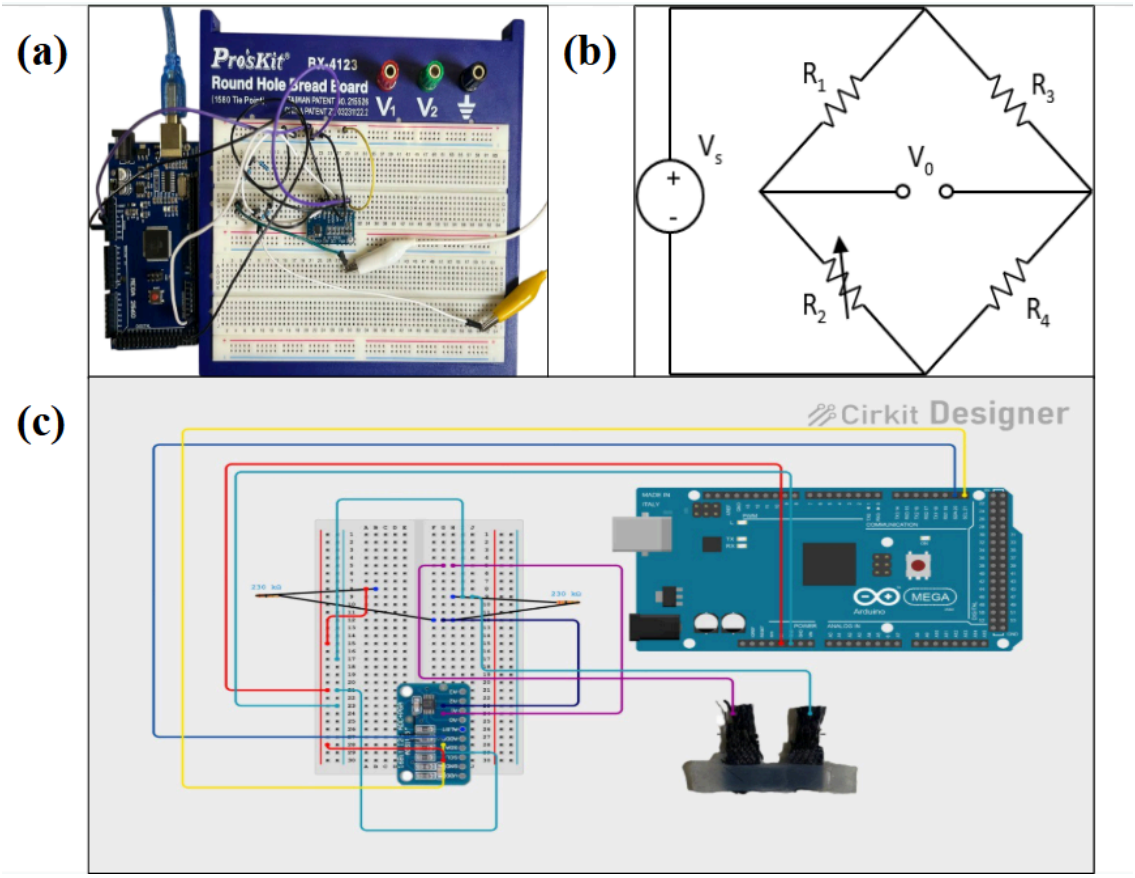

Figure 18. (a) Experimental setup of Arduino Mega 2560 with a breadboard; (b) Schematic diagram of a Wheatstone bridge circuit; c) Wiring diagram of strain sensor, Wheatstone bridge and Arduino Mega 2560 connections to breadboard.

The problem was the inability to measure the change in resistance, because Arduino can only read voltage. It usually suits pressure sensors, but not the resistance ones. So a constant current source was connected to the circuit. But still, the system did not give the output that was needed. To solve this problem, a Wheatstone bridge was added to the circuit. As shown in **Figure 18b,** 3 resistors with constant resistance were connected to the circuit, and R2 is a strain sensor with changing resistance.

After that, tests were conducted to find a correlation between the sensor resistance and the angular change in the robot's position. But before that, to understand whether the values shown by the sensor are stable or not, the robot was moved under a sinusoidal motion, starting from the starting position and then changing to one of the 6 positions after returning to the initial one. The results of this can be seen in the figure below:

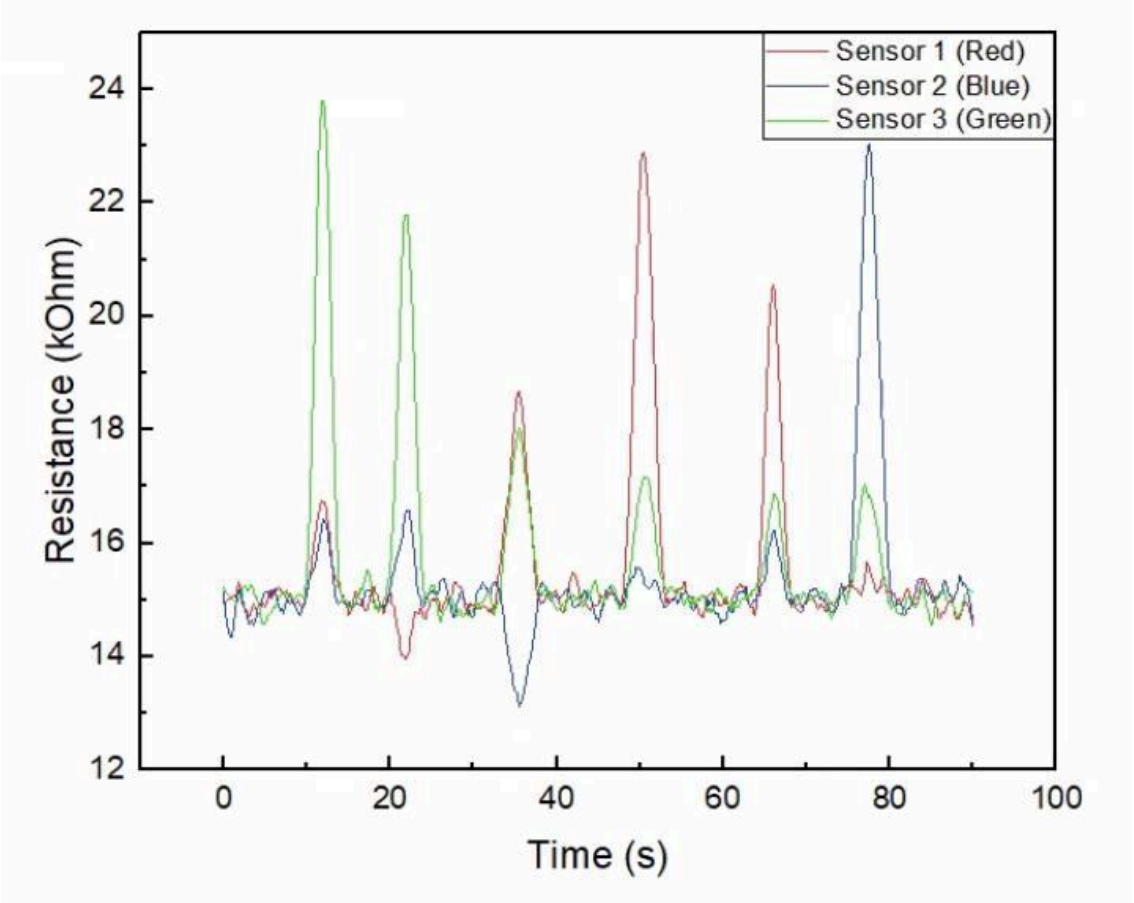

Figure 19. Resistance over Time graph in sinusoidal motion

From the graph, it can be seen that sensor values are stable and changing according to the strain that occurs due to the movement of the robot.

The next part is the angle correlation. The measured angle of the robot can be seen in **Figure 20a**. It varies from -65 degrees to +65 degrees, where -65 is the maximum displacement of the robot to the left and +65 is the maximum displacement to the right. **Figure 20b** shows the relationship between the resistance of the sensors and the robot's angle.

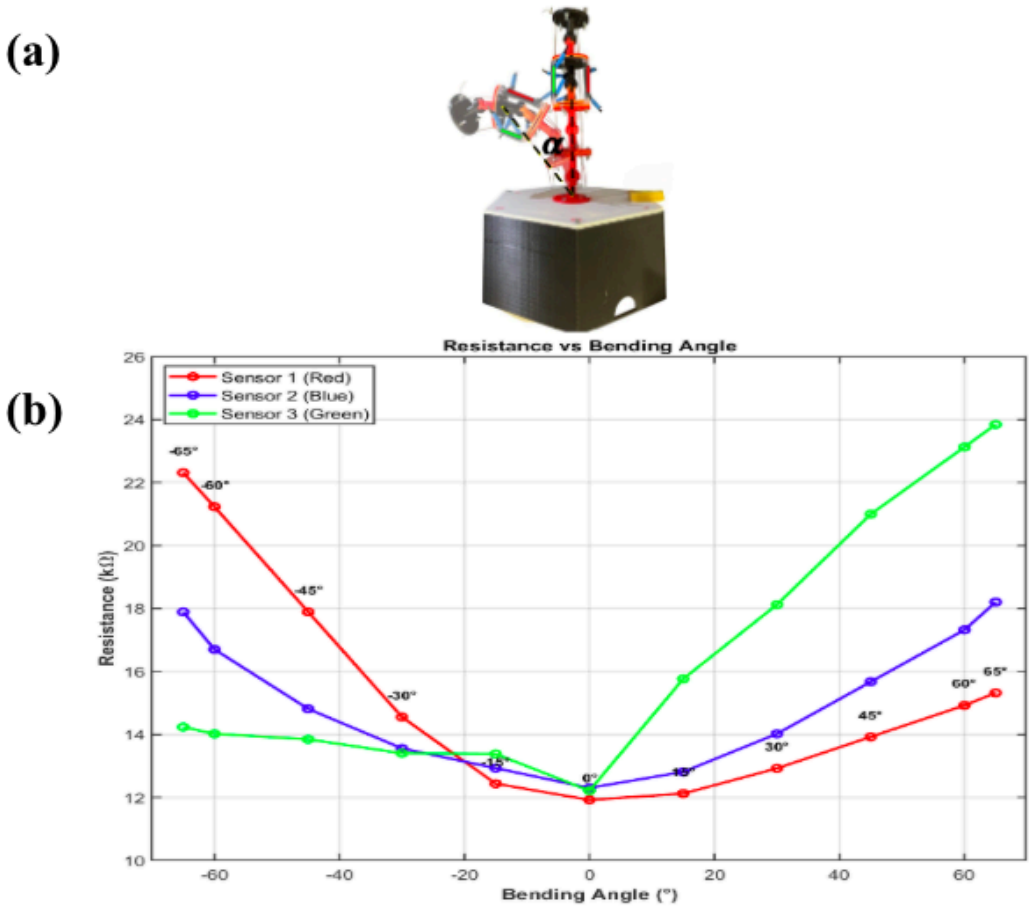

Figure 20. a) The Calculated Robot Angle; b) Resistance over Bending angle graph in sinusoidal motion.

During the joystick-controlled manipulation of the robot, motor position readings were collected and correlated with the corresponding angle of rotation values. **Figure 21** illustrates this correlation specifically for movements where the angle of bending input was set to 45 degrees. All three motors demonstrate cyclical behavior approximating a sinusoidal pattern over the 360-degree cycle, which validates the expectations derived from the robot's kinematic analysis.

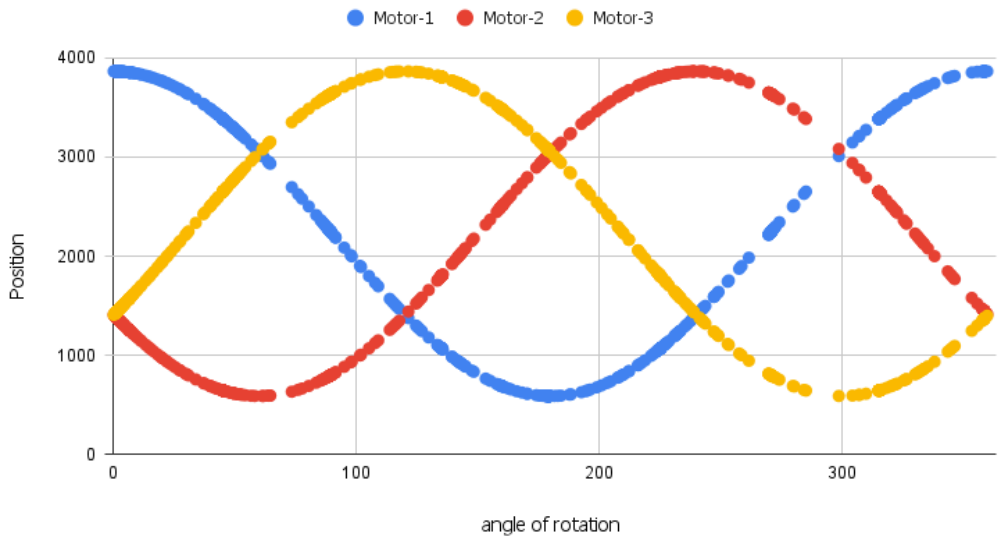

Figure 21. Motor position vs the robot's angle of rotation (α) at a specified angle of bending (θ = 45).

A demonstration video showcasing the continuum robot's movement and motor control implementation can be found at the following link:

Continuum Robot (second video)

5. Recommendations

Exploring the use of machine learning for sensor data analysis could allow the automated determination of motor inputs from sensor feedback. Currently, strain sensors are integrated only to the robot’s single section. Therefore, future improvements could involve distributed integration of strain sensors over multiple sections of the robot. This could potentially enhance the consistency of bending actions, minimize unwanted twisting, and refine the relationship established between motor commands and sensor responses. The implemented kinematic model is based upon simplified assumptions, namely uniform bending curvature and the disregard of gravitational loading effects. Furthermore, it assumes idealized sensor fabrication, a condition difficult to achieve due to inherent manufacturing tolerances and variability. Therefore, an empirical calibration approach is recommended, utilizing extensive data collection to accurately map motor control inputs to the robot's resultant motion, thereby compensating for these modeling and fabrication imperfections. The robot's current design permits only bending actuation, the extent of which is limited by the nature of its ball-and-socket joints. Incorporating additional degrees of freedom, such as compression, extension, and torsion, through an alternative design could improve spatial freedom and overall utility. Further exploration of different robot height and size parameters is recommended. Such variations could potentially improve both the robot's range of motion and its overall structural integrity. This development would require a parallel investigation into the fabrication and implementation of strain sensors sufficiently sized for these different scales. The potential to develop sensor-integrated CDCRs at various sizes could broaden their range of applications.

6. Conclusion

During this project a cable driven continuum robot was initially designed based on a ball and socket joint system, fabricated using 3D printing using PLA and later mechanical tests took place. Alongside the CDCR the strain sensors were fabricated as mentioned earlier using multi-walled carbon nanotubes as the main component that was later wrapped into between Ecoflex, thus ensuring both structural integrity and required properties of the strain sensor. Following the fabrication stage the testing of the strain sensors showed great results of linearity (0.981), low hysteresis(1.59%) and sensitivity (4.12) and overall ability to withstand long term stretch without structural damages to the Ecoflex base.

Following earlier works that mentioned the benefits of the modular approach in robotics, the platforms were designed and fabricated similarly, as a result the whole structure was much sturdy and was able to be stable and withstand the torque from motors and the overall structure was built in such a way that easy access to motors or any other parts was available. During the testing stage of sensors it was revealed that the third section from the bottom offered the most strain response, thus the sensors were mounted solely on the mentioned section, as a result the resistance data quality and strain sensing accuracy was significantly enhanced. which was later validated by both manually and with the utilization of the Dynamixel MX-28 motors and the correlation between strain and resistance was validated once more.

Albeit the testing was not as smooth as it could have been, as non-uniform bending behavior was observed during actuation which was later identified occurring due to sensor absence on other sections and the freedom of rotation present in the ball and socket joint structure. In addition, there were discrepancies between the theoretical and actual deformation models, which subsequently means that further more research and proper in-depth examinations will be required.

AI on the other hand was utilized in both the cases of data enhancement and circuit connection during the data obtaining stage. With the help of AI the data obtained had been cleared of noise and other obstacles and the AI was able to assist the connection to the Arduino motherboard of the sensors.

For the future it will be recommended that the AI focus will be towards autonomous control and data acquisition and based on the potential utilization of the CDCR redesigning the CDCR ball and socket joint, possibly allowing a higher degree of freedom in rotation if required. Investigating alternative sensor placement could also benefit the structure and offer great data acquisition, on top of that possibly adding more sensors effectively without the sensors interfering with each other could improve the current design. Finally, researching on innovative designs and advanced materials and implementing them according to the needs in the future will allow to implement the given CDCR in not only constrained and complex environments such as in rescue missions but also in minimally invasive surgeries and space explorations.

7. References

[1] I. D. Walker, “Continuous backbone continuum robot manipulators,” *IRSN Robot*, vol. 2013, pp. 1-19, 2013.

[2] Z. Liu, X. Zhang, Z. Cai, H. Peng and Z. Wu, “Real-Time Dynamics of Cable-Driven Continuum Robots Considering the Cable Constraint and Friction Effect,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6235-6242, 2021.

[3] A. Yeshmukhametov, K. Koganezawa, and Y. Yamamoto, “A novel discrete wire-driven continuum robot arm with passive sliding disc: Design, kinematics and passive tension control,” *Robotics*, vol. 8, no. 3, p. 51, 2019.

[4] I.D. Walker, C. Carreras, R. McDonnell, and G. Grimes, “Extension versus bending for continuum robots,” *International Journal of Advanced Robotic Systems*, vol. 3, no. 2, p. 26, 2006.

[5] Y. Yamauchi, Y. Ambe, H. Nagano, M. Konyo, Y. Bando, E. Ito, S. Arnold, K. Yamazaki, K. Itoyama, T. Okatani, and H.G. Okuno, “Development of a continuum robot enhanced with distributed sensors for search and rescue,” *Robomech Journal*, vol. 9, no. 1, p.8, 2022.

[6] D. Nahar, P.M. Yanik, and I.D. Walker, “Robot tendrils: Long, thin continuum robots for inspection in space operations,” *In 2017 IEEE aerospace conference*, pp. 1-8, 2017.

[7] W. Shen, J. He, G. Yang, X. Kong, H. Bai, and Z. Fang, “Shape Sensing and Kinematic Control of a Cable-Driven Continuum Robot Based on Stretchable Capacitive Sensors,” *Sensors*, vol. 24, no. 11, p. 3385, 2024.

[8] I. Queiroz, M. Vieira, H. Monteiro, M. Cerqueira, C. Viturino, and L. Silva, “Development and Validation of a 3-D Printed Tendon-Driven Flexible Continuum Robot,” *In 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE),* pp. 135-140, 2023.

[9] S. Ryu, P. Lee, J.B. Chou, R. Xu, R., Zhao, A.J. Hart, and S.G. Kim, “Extremely Elastic Wearable Carbon Nanotube Fiber Strain Sensor for Monitoring of Human Motion,” *ACS Nano,* vol. 9, pp. 5929–5936, 2015.

[10] J. Shintake, E. Piskarev, S.H. Jeong, and D. Floreano, “Ultrastretchable Strain Sensors Using Carbon Black-Filled Elastomer Composites and Comparison of Capacitive Versus Resistive Sensors,” *Adv. Mater. Technol.*, vol. 3, 2017.

[11] Z. Zhang, Q. Liao, X. Zhang, G. Zhang, P. Li, S. Lu, S. Liu, and Y. Zhang, “Highly efficient piezotronic strain sensors with symmetrical Schottky contacts on the monopolar surface of ZnO nanobelts,” *Nanoscale,* vol. 7, pp. 1796–1801, 2015.

[12] F. Yi, L. Lin, S. Niu, P.K. Yang, Z. Wang, J. Chen, Y. Zhou, Y. Zi, J. Wang, Q. Liao, Y. Zhang, and Z.L Wang, “Stretchable-Rubber-Based Triboelectric Nanogenerator and Its Application as Self-Powered Body Motion Sensors,” *Adv. Funct. Mater.*, vol. 25, pp. 3688–3696, 2015.

[13] R. Das, K. Ghaffarzadeh, and X. He, “Flexible, Printed and Organic Electronics 2019–2029: Forecasts, Players & Opportunities,” *IDTechEx: Cambridge, UK,* 2018.

[14] S. Qian, T. Zheng, W. Shen, and H. Bai, ‘Feedback control of Cable-Driven continuum robot based on flexible perception,” *In 2023 2nd International Symposium on Control Engineering and Robotics (ISCER)*, pp. 64-69, 2023.

[15] A. Zolfagharian, M. Bodaghi, A. Winter, and D. Nahavandi, “Non-Assembly Spherical Joints 3D-Printed for Soft Robotic Applications,” *In 2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE),* pp. 1-6, 2021.

[16] S.K. Mahapatra, and A. Ghosal, “3D printed cable-driven continuum robots with generally routed cables: modeling and experiments,” *arXiv preprint arXiv:2003.04593,* 2020.

[17] S. Zhou, Y. Li, Q. Wang, and Z. Lyu, “Integrated Actuation and Sensing: Toward Intelligent Soft Robots,” *Cyborg and Bionic Systems*, vol. 5, p. 0105, 2024.

[18] H. Souri, H. Banerjee, A. Jusufi, N. Radacsi, A.A. Stokes, I. Park, M. Sitti, and M. Amjadi, "Wearable and stretchable strain sensors: materials, sensing mechanisms, and applications," *Advanced Intelligent Systems,* vol. 2, no. 8, p. 2000039, 2020.

[19] J. Rivnay, S. Inal, A. Salleo, R.M. Owens, M. Berggren, and G.G. Malliaras, "Organic electrochemical transistors," *Nature Reviews Materials*, vol. 3, no. 2, pp. 1-14, 2018.

[20] L. Groenendaal, F. Jonas, D. Freitag, H. Pielartzik, and J.R. Reynolds, "Poly (3, 4‐ethylenedioxythiophene) and its derivatives: past, present, and future," *Advanced materials*, vol. 12, no. 7, pp. 481-494, 2000

[21] J. Zhang, X. Chang, and J. Zhang, "Conducting polymer-based flexible strain sensors: A review," *Sensors*, vol. 19, no. 3, p. 599, 2019.

[22] A.K. Geim, and K.S. Novoselov, "The rise of graphene," *Nature Materials*, vol. 6, no. 3, pp. 183–191, 2007.

[23] D.S. Hecht, L. Hu, and G. Irvin, "Emerging transparent electrodes based on thin films of carbon nanotubes, graphene, and metallic nanostructures," *Advanced Materials*, vol. 23, no. 13, pp. 1482–1513, 2011.

[24] F. Bonaccorso, Z. Sun, T. Hasan, and A.C. Ferrari, "Graphene photonics and optoelectronics," *Nature Photonics*, vol. 4, no. 9, pp. 611–622, 2010.

[25] A.R. Rathmell, and B.J. Wiley, "The synthesis and coating of long, thin copper nanowires to make flexible, transparent conducting films on plastic substrates," *Advanced Materials*, vol. 23, no. 41, pp. 4798–4803, 2011.

[26] J.N. Lee, C. Park, and G.M. Whitesides, "Solvent compatibility of poly(dimethylsiloxane)-based microfluidic devices," *Analytical Chemistry*, vol. 75, no. 23, pp. 6544–6554, 2003.

[27] Z. Ma, M. Kotaki, and S. Ramakrishna, "Electrospun cellulose nanofiber as affinity membrane," *Journal of Membrane Science*, vol. 265, no. 1-2, pp. 115–123, 2005.

[28] D.M. Vogt, and R.J. Wood, "Design and characterization of a soft multi-axis force sensor using embedded microfluidic channels," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 396-406, 2018.

[29] H. Liu, Q. Li, S. Zhang, R. Yin, X. Liu, Y. He, K. Dai, C. Shan, J. Guo, C. Liu, and C. Shen, "Electrically conductive polymer composites for smart flexible strain sensors: a critical review," *Journal of Materials Chemistry C*, vol. 6, no. 45, pp. 12121-12141, 2018

[30] M. Amjadi, K.-U Kyung, I. Park, and M. Sitti, "Stretchable, skin-mountable, and wearable strain sensors and their potential applications: A review," *Advanced Functional Materials*, vol. 26, no. 11, pp. 1678–1698, 2016.

[31] I. Akhtar, and S.-H. Chang, "Highly aligned carbon nanotubes and their sensor applications," *Nanoscale*, vol. 12, no. 41, pp. 21447–21458, 2020.

[32] C. Wang, J. Zhao, C. Ma, J. Sun, L. Tian, X. Li, F. Li, X. Han, C. Liu, C., Shen, and L. Dong, "Detection of non-joint areas, tiny strain and anti-interference voice recognition by micro-cracked metal thin film," *Nano Energy*, vol. 34, pp. 578-585, 2017.

[33] Y. Li, T. He, L. Shi, R. Wang, and J. Sun, "Strain Sensor with Both a Wide Sensing Range and High Sensitivity Based on Braided Graphene Belts," *ACS Applied Materials & Interfaces*, vol. 12, no. 15, pp. 17691-17698, 2020.

[34] S. Zhu, J.H. So, R. Mays, S. Desai, W.R. Barnes, B. Pourdeyhimi, and M.D. Dickey, "Ultrastretchable fibers with metallic conductivity using a liquid metal alloy core," *Advanced Functional Materials*, vol. 23, no. 18, pp.2308-2314, 2013.

[35] S.Y. Jeong, J.U. Lee, S.M. Hong, C.W. Lee, S.H. Hwang, S.C. Cho, and B.S. Shin, "Highly skin-conformal laser-induced graphene-based human motion monitoring sensor," *Nanomaterials*, vol. 11, no. 4, p. 951, 2021.

[36] A.N. Liang, R. Stewart, and N. Bryan-Kinns, "Analysis of sensitivity, linearity, hysteresis, responsiveness, and fatigue of textile knit stretch sensors," *Sensors (Switzerland),* vol. 19, no. 16, p. 3618, 2019.

[37] Y. Ko, J.S. Kim, C.C. Vu, and J. Kim, "Ultrasensitive strain sensor based on pre-generated crack networks using Ag nanoparticles/single-walled carbon nanotube (SWCNT) hybrid fillers and a polyester woven elastic band," *Sensors*, vol. 21, no. 7, p. 2531, 2021.

[38] X. Xie, H. Huang, J. Zhu, J. Yu, Y. Wang, and Z. Hu, “A spirally layered carbon nanotube-graphene/polyurethane composite yarn for highly sensitive and stretchable strain sensor,” *Composites Part A: Applied Science and Manufacturing*, vol. 135, p. 105932, 2020.

[39] A.M. Soomro, M.A.U. Khalid, I. Shah, S. wan Kim, Y.S. Kim, and K.H. Choi, “Highly stable soft strain sensor based on Gly-KCl filled sinusoidal fluidic channel for wearable and water-proof robotic applications,” *Smart Materials and Structures,* vol. 29,no. 2, p. 025011, 2020

[40] Y. Lin, S. Liu, S. Chen, Y. Wei, X. Dong, and L. Liu, “A highly stretchable and sensitive strain sensor based on graphene–elastomer composites with a novel double-interconnected network,” *Journal of Materials Chemistry C*, vol. 4, no. 26, pp. 6345-6352, 2016.

[41] M. Wang, K. Zhang, X.X. Dai, Y. Li, J. Guo, H. Liu, G.H. Li, Y.J, Tan, J.B. Zeng, and Z. Guo, “Enhanced electrical conductivity and piezoresistive sensing in multi-wall carbon nanotubes/polydimethylsiloxane nanocomposites via the construction of a self-segregated structure,” *Nanoscale*, vol. 9, no. 31, pp. 11017-11026, 2017.

[42] F. Han, J. Li, S. Zhao, Y. Zhang, W. Huang, G. Zhang, R. Sun, and C.P. Wong, “A crack-based nickel@ graphene-wrapped polyurethane sponge ternary hybrid obtained by electrodeposition for highly sensitive wearable strain sensors,” *Journal of Materials Chemistry C,* vol. 5, no. 39, pp. 10167-10175, 2017.

[43] X. Li, K. H. Koh, M. Farhan, and K.W.C. Lai, “An ultra flexible polyurethane yarn-based wearable strain sensor with a polydimethylsiloxane infiltrated multilayer sheath for smart textiles,” *Nanoscale*, vol. 12, no. 6, pp. 4110-4118, 2020.

[44] S. Sun, Y. Liu, X.Chang, Y. Jian, D. Wang, C. Tang, S. He, M. Wang, L. Guo, and Y. Gao, “A wearable, waterproof, and highly sensitive strain sensor based on three-dimensional graphene/carbon black/Ni sponge for wirelessly monitoring human motions,” *Journal of Materials Chemistry C*, vol. 8, no. 6, pp. 2074-2085, 2020.

[45] P. Wang, W. Wei, Z. Li, W. Duan, H. Han, and Q. Xie, “A superhydrophobic fluorinated PDMS composite as a wearable strain sensor with excellent mechanical robustness and liquid impalement resistance,” *Journal of Materials Chemistry A*, vol. 8, no. 6, pp. 3509-3516, 2020.

[46] M. Aakyiir, J.-A. Oh, S. Araby, Q. Zheng, M. Naeem, J. Ma, P. Adu, L. Zhang, and Y.-W. Mai, “Combining hydrophilic mxene nanosheets and hydrophobic carbon nanotubes for mechanically resilient and electrically conductive elastomer nanocomposites,” *Composites Science and Technology,* vol. 214, p. 108997, 2021.

[47] H. L. Liu, D. J. Kou, D. F. Yu, K. Y. Ge and Y. M. Hu, "Kinematics and Dynamics Modeling and Analysis of Line-Driven Continuum Robot," *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE),* pp. 97-100, 2020.

[48] R. Onose, and H. Sawada, “A ball-jointed tendon-driven continuum robot with multi-directional operability for grasping objects,” *ROBOMECH Journal*, vol. 11, no. 1, p. 4, 2024.

[49] Z. Zhang, S. Tang, W. Fan, Y. Xun, H. Wang, and G. Chen, “Design and analysis of hybrid-driven origami continuum robots with extensible and stiffness-tunable sections,” *Mechanism and Machine Theory,* vol. 169, p. 104607, 2022.

Appendix

A.1 First Section

Arduino code for the joystick:

```
#include <Arduino.h>
#include <math.h>

// --- Configuration ---
const int joyXPin = A0;        // Analog pin for Joystick X-axis
const int joyYPin = A1;        // Analog pin for Joystick Y-axis

// Joystick characteristics
const int joyCenter = 512;
const int joyDeadzone = 50;

// Define the constant theta_m value
const float theta_m = 45.0;

const long serialBaudRate = 9600;

void setup() {
  Serial.begin(serialBaudRate);
  delay(100);
  Serial.println("Arduino Theta_m (Fixed) and Alpha Sender Ready");
}

void loop() {
  int joyXVal = analogRead(joyXPin);
  int joyYVal = analogRead(joyYPin);
```

```
  float x_coord = (float)(joyXVal - joyCenter);
  float y_coord = (float)(joyYVal - joyCenter);

  float alpha_rad = atan2(y_coord, x_coord);

  // Convert radians to degrees [0, 360)
  float alpha_deg = alpha_rad * 180.0 / PI;
  if (alpha_deg < 0) {
    alpha_deg += 360.0;
  }

  float magnitude = sqrt(x_coord * x_coord + y_coord * y_coord);

  if (magnitude > joyDeadzone) {
    Serial.print(theta_m, 2);
    Serial.print(",");
    Serial.println(alpha_deg, 2);
  } else {

  delay(50);
```

The motor controller code:

```
import os
import serial
import time
import math
import sys
import signal
```

```
from dynamixel_sdk import *

# --- Configuration ---


ARDUINO_SERIAL_PORT = 'COM8'

ARDUINO_BAUD_RATE = 9600

# Dynamixel SDK Settings

DXL_DEVICENAME              = 'COM9'

DXL_PROTOCOL_VERSION        = 1.0

DXL_BAUDRATE                = 57600


# Dynamixel IDs

DXL_ID_1                    = 1

DXL_ID_2                    = 2

DXL_ID_3                    = 3

DXL_IDS                     = [DXL_ID_1, DXL_ID_2, DXL_ID_3]


ADDR_MX_TORQUE_ENABLE       = 24

ADDR_MX_GOAL_POSITION       = 30

ADDR_MX_PRESENT_POSITION    = 36


# Dynamixel Torque Settings

TORQUE_ENABLE               = 1

TORQUE_DISABLE              = 0


# Dynamixel Position Limits (used in calculation function)

DYNAMIXEL_MIN_POS           = 0

DYNAMIXEL_MAX_POS           = 4095


# --- Calculation Constants & Functions ---
```

```
H_SPACER = 25

R_TENDON = 22.5

L_MAX    = 200

C1_MULT  = 4

C2_DIV   = 14

C3_CONV  = 2048 / math.pi

C4_OFFSET= 2048


def calculate_t_values(theta_m, alpha):

    """Calculates intermediate t values based on angles."""

    theta_m_clamped = min(theta_m, 89.99)

    theta = (90 - theta_m_clamped) / 4

    if theta < 0: theta = 0

    theta_rad = math.radians(theta)

    alpha_rad = math.radians(alpha)

    sin_theta_rad = math.sin(theta_rad)

    if abs(sin_theta_rad) < 1e-9:

        t_val = 2 * H_SPACER

        return round(t_val, 3), round(t_val, 3), round(t_val, 3)

    t_1 = 2 * (H_SPACER * math.cos(theta_rad) - math.cos(alpha_rad)
* R_TENDON * sin_theta_rad)

    t_2 = 2 * (H_SPACER * math.cos(theta_rad) -
math.cos(math.radians(alpha + 120)) * R_TENDON * sin_theta_rad)

    t_3 = 2 * (H_SPACER * math.cos(theta_rad) -
math.cos(math.radians(alpha + 240)) * R_TENDON * sin_theta_rad)

    return round(t_1, 3), round(t_2, 3), round(t_3, 3)


def calculate_n_values(t_1, t_2, t_3):

    """Calculates final Dynamixel position values (n) from t
values."""
```

```
    if abs(C2_DIV) < 1e-9:
        print("Error: Division by zero in n-value calculation
(C2_DIV is zero).")
        return int(C4_OFFSET), int(C4_OFFSET), int(C4_OFFSET)
    m_1 = L_MAX - C1_MULT * t_1
    m_2 = L_MAX - C1_MULT * t_2
    m_3 = L_MAX - C1_MULT * t_3
    n_1 = int((m_1 / C2_DIV) * C3_CONV + C4_OFFSET)
    n_2 = int((m_2 / C2_DIV) * C3_CONV + C4_OFFSET)
    n_3 = int((m_3 / C2_DIV) * C3_CONV + C4_OFFSET)
    n_1 = max(DYNAMIXEL_MIN_POS, min(DYNAMIXEL_MAX_POS, n_1))
    n_2 = max(DYNAMIXEL_MIN_POS, min(DYNAMIXEL_MAX_POS, n_2))
    n_3 = max(DYNAMIXEL_MIN_POS, min(DYNAMIXEL_MAX_POS, n_3))
    return n_1, n_2, n_3
# --- End Calculation Functions ---
# --- Global Variables ---
ser_arduino = None # Serial port for Arduino
portHandler = None # Dynamixel SDK Port Handler
packetHandler = None # Dynamixel SDK Packet Handler
running = True     # Flag to control the main loop
incoming_buffer = b'' # Buffer for storing raw bytes from Arduino
serial


# --- Graceful Exit Handler ---
def signal_handler(sig, frame):
    """Handles Ctrl+C interruption to ensure cleanup."""
    global running
    print('\nCtrl+C detected. Preparing to shut down...')
    running = False
signal.signal(signal.SIGINT, signal_handler) # Register the handler
```

```
# --- Function to pause and keep window open ---
def pause_exit(message=""):
    """Prints a message and waits for Enter key before exiting."""
    if message:
        print(message)
    input("--- Press Enter to exit ---")
    sys.exit(1)

# --- Main Program Execution ---
print("--- Combined Arduino Reader & Dynamixel Controller ---")
print("--- Script Starting ---")

try:
    # 1. Initialize Arduino Serial Port
    print(f"--- [SETUP STEP 1/5] Attempting to connect to Arduino on {ARDUINO_SERIAL_PORT}... ---")
    try:
        ser_arduino = serial.Serial(
            port=ARDUINO_SERIAL_PORT,
            baudrate=ARDUINO_BAUD_RATE,
            timeout=0.02
        )
        time.sleep(2)
        ser_arduino.reset_input_buffer()
        print(f"--- [SETUP STEP 1/5] Successfully connected to Arduino on {ser_arduino.name} ---")
    except serial.SerialException as e:
        pause_exit(f"--- FATAL ERROR (Arduino Connect): Could not open port {ARDUINO_SERIAL_PORT}: {e} ---")
```

```
    except Exception as e:
        pause_exit(f"--- FATAL ERROR (Arduino Connect): Unexpected error: {e} ---")

    # 2. Initialize Dynamixel SDK PortHandler
    print(f"--- [SETUP STEP 2/5] Initializing Dynamixel SDK PortHandler for {DXL_DEVICENAME}... ---")
    try:
        portHandler = PortHandler(DXL_DEVICENAME)
        print(f"--- [SETUP STEP 2/5] PortHandler initialized. ---")
    except Exception as e:
        pause_exit(f"--- FATAL ERROR (DXL PortHandler Init): {e} ---")

    # 3. Initialize Dynamixel SDK PacketHandler
    print(f"--- [SETUP STEP 3/5] Initializing Dynamixel SDK PacketHandler (Protocol {DXL_PROTOCOL_VERSION})... ---")
    try:
        packetHandler = PacketHandler(DXL_PROTOCOL_VERSION)
        print(f"--- [SETUP STEP 3/5] PacketHandler initialized. ---")
    except Exception as e:
        pause_exit(f"--- FATAL ERROR (DXL PacketHandler Init): {e} ---")

    # 4. Open Dynamixel Port
    print(f"--- [SETUP STEP 4/5] Opening Dynamixel port {DXL_DEVICENAME}... ---")
    if portHandler.openPort():
        print(f"--- [SETUP STEP 4/5] Succeeded to open the Dynamixel port ---")
    else:
```

```
        portHandler.closePort()
        time.sleep(0.5)
        if portHandler.openPort():
             print(f"--- [SETUP STEP 4/5] Succeeded to open the
Dynamixel port on second attempt ---")
        else:
            pause_exit(f"--- FATAL ERROR (DXL Open Port): Failed to
open port {DXL_DEVICENAME}. Check connection and ensure it's not in
use. ---")


    print(f"--- [SETUP STEP 5/5] Setting Dynamixel baudrate to
{DXL_BAUDRATE}... ---")
    if portHandler.setBaudRate(DXL_BAUDRATE):
        print(f"--- [SETUP STEP 5/5] Succeeded to change the
Dynamixel baudrate ---")
    else:
                portHandler.closePort() # Close port before exiting
        pause_exit(f"--- FATAL ERROR (DXL Set Baudrate): Failed.
Verify motor baudrate matches {DXL_BAUDRATE} and check port
connection/permissions. ---")
    print("--- [SETUP FINAL] Enabling Torque for Dynamixels...
---")
    all_torque_enabled = True
    for motor_id in DXL_IDS:
        print(f"    Attempting Torque Enable for ID:
{motor_id}...")
        dxl_comm_result, dxl_error =
packetHandler.write1ByteTxRx(portHandler, motor_id,
ADDR_MX_TORQUE_ENABLE, TORQUE_ENABLE)
        time.sleep(0.05)
        if dxl_comm_result != COMM_SUCCESS:
            error_msg =
packetHandler.getTxRxResult(dxl_comm_result)
```

```
            print(f"    ERROR: ID {motor_id} Torque Enable Comm
Failed: {error_msg}")
            all_torque_enabled = False
        elif dxl_error != 0:
            error_msg = packetHandler.getRxPacketError(dxl_error)
            print(f"    ERROR: ID {motor_id} Torque Enable Packet
Error: {error_msg}")
                        all_torque_enabled = False
        else:
            print(f"    SUCCESS: ID {motor_id} torque enabled.")


    if not all_torque_enabled:
        portHandler.closePort() # Close port before exiting
        pause_exit("--- FATAL ERROR: Failed to enable torque on one
or more Dynamixels. Check IDs, Baudrate, Power, Wiring. ---")


    print("\n--- SETUP COMPLETE. Starting main loop. Reading
Arduino and controlling Dynamixels... ---")
    print("--- Press Ctrl+C to exit. ---")
    # input("--- Press Enter to start loop (DEBUG PAUSE) ---") #
Optional pause


    # >>> MAIN LOOP <<<
    while running:
        processed_a_line = False
        target_n1, target_n2, target_n3 = None, None, None # Store
latest targets


        # --- Read and Process Arduino Data ---
        try:
            if not ser_arduino.is_open:
```

```
                print("\n--- WARNING: Arduino serial port closed
unexpectedly! Shutting down. ---")
                running = False
                break # Exit loop cleanly

            bytes_to_read = ser_arduino.in_waiting
            if bytes_to_read > 0:
                raw_data = ser_arduino.read(bytes_to_read)
                incoming_buffer += raw_data

            while b'\n' in incoming_buffer and running:
                line_bytes, incoming_buffer =
incoming_buffer.split(b'\n', 1)
                line_bytes = line_bytes.strip(b'\r')

                try:
                    line_str = line_bytes.decode('utf-8').strip()
                except UnicodeDecodeError:
                    # print(f"Warning: Decoding error on data:
{line_bytes!r}") # Optional debug
                    continue # Skip corrupted line

                if line_str:
                    parts = line_str.split(',')
                    if len(parts) == 2:
                        try:
                            theta_m = float(parts[0])
                            alpha = float(parts[1])
                            processed_a_line = True
```

```
                            # Perform calculations
                            t1, t2, t3 =
calculate_t_values(theta_m, alpha)
                            target_n1, target_n2, target_n3 =
calculate_n_values(t1, t2, t3)


                            # === Set Dynamixel Goal Positions ===
                            write_error = False
                            for motor_id, goal_pos in zip(DXL_IDS,
[target_n1, target_n2, target_n3]):
                                dxl_comm_result, dxl_error =
packetHandler.write2ByteTxRx(portHandler, motor_id,
ADDR_MX_GOAL_POSITION, goal_pos)
                                if dxl_comm_result != COMM_SUCCESS
or dxl_error != 0:
                                    # Print error only once per
loop iteration if it occurs
                                    if not write_error:
                                        print(f"\nDXL WritePos
Error - ID:{motor_id}
Comm:{packetHandler.getTxRxResult(dxl_comm_result)}
Packet:{packetHandler.getRxPacketError(dxl_error)}", end='')
                                    write_error = True
                                # time.sleep(0.001) # Tiny delay if
needed, often not required


                            # === Display Status ===
                            if not write_error: # Only print status
if writes were okay
                                status_line = (f"Recv:
th={theta_m:6.2f}, al={alpha:7.2f} | "
                                               f"Goal:
n1={target_n1:4d}, n2={target_n2:4d}, n3={target_n3:4d} ")
                                print(status_line, end='\r')
```

```
                        except ValueError:


                            pass


        except serial.SerialException as e:
            print(f"\n--- ARDUINO SERIAL ERROR during
read/process: {e}. Shutting down. ---")
            running = False
            break
        except Exception as e:
            print(f"\n--- UNEXPECTED ERROR in main loop:
{type(e).__name__}: {e}. Attempting to continue... ---")
            time.sleep(0.1)


        # --- Loop Speed Control ---
        if not processed_a_line:
            time.sleep(0.005)


except KeyboardInterrupt:
    print('\n--- KeyboardInterrupt caught by main try block.
Preparing to shut down. ---')
    running = False
finally:
    # --- Cleanup Resources ---
    print("\n--- Exiting main loop. Cleaning up resources... ---")
    if packetHandler is not None and portHandler is not None and
portHandler.is_open:
        print("--- Disabling Dynamixel torque... ---")
        for motor_id in DXL_IDS:
```

```
            print(f"    Disabling torque for ID: {motor_id}...")
            packetHandler.write1ByteTxRx(portHandler, motor_id,
ADDR_MX_TORQUE_ENABLE, TORQUE_DISABLE)
            time.sleep(0.05) # Small delay
        print("--- Torque disable commands sent. ---")
    if portHandler is not None and portHandler.is_open:
        print("--- Closing Dynamixel port... ---")
        portHandler.closePort(
    if ser_arduino and ser_arduino.is_open:
        print("--- Closing Arduino serial port... ---")
        ser_arduino.close()

    print("--- Shutdown complete. ---")
    input("--- Press Enter to close window ---")
```