



MECH0064 MSc Group Design Project

**Compact Continuum Robotic
Manipulator Platform**

Group members:

Zehao Ye (23119333) Yuhao Zhu (23041703)

Zehao Ye (23119333) Zehao Ye (23119333)

Zehao Ye (23119333) Zehao Ye (23119333)

Supervised by Dr Reza Haqshenas

Abstract

This is the Abstract of the final report.

AAAAA bbb

test for github

Key Words: **Continuum Robotic e.g.**

Contents

	Page
Abstract	I
List of Figures	II
List of Tables	III
1. Introduction	1
1.1 Background	1
1.2 Motivation	1
1.3 Introduction	1
2. Literature Review	2
2.1 Manipulators in Biology Field	2
2.2 Types of Manipulator	3
2.3 Types of Continuum Robots	4
2.3.1 Tendon-Driven Robots	4
2.3.2 Fishbone Robots	5
2.3.3 Concentric Tube Continuum Robots	6
3. Design	8
3.1 Structural Design	8
3.2 Designation and Parameter Definition of Manipulator	8
3.3 Material Selection	9
3.4 Methodology	12
3.4.1 Forward Kinematics	12
3.4.2 Inverse Kinematics	18
3.5 Electronic Control	24
3.5.1 Manipulator Actuation Control	24
3.5.2 Parameter input by Serial Monitor	28

3.5.3	Verification of MPU 6050	29
3.5.4	Using HC-SR04 to measure distance	30
3.5.5	SSD 1306 OLED	32
4.	Result and Discussion	33
4.1	Error Analysis of Cross-Shaped Sheets	33
4.2	Workspace Analysis of Manipulator	34
4.3	Inverse Kinematics	39
4.3.1	Singular Posture Solution	39
4.3.2	Trajectory Replication	41
4.3.3	The improvement about Inverse kinematics algorithm	42
4.4	Electroinc Control	43
5.	Conclusion	45

References	i
-------------------	----------

A. The Appendix of Tables	ix
----------------------------------	-----------

B. Test	xiii
----------------	-------------

List of Figures

1	The three tendons continuum robot with one segment	5
2	The cable-driven fish bone continuum robot	6
3	An example concentric tube continuum robot	7
4	The main components of the proposed manipulator and cables arrangement	8
5	The kinematics model of manipulator in the initial position	13
6	The kinematics model of manipulator with respective bending modules .	14
7	The flow chart of the FABRIKc algorithm	19
8	The circuit layout of Arduino step motor control system	26
9	The flow chart of Arduino step motor control programme	27
10	The code logic of input parameters and MPU6050	29
11	The MPU6050 chip with ancillary Arduino circuit	30
12	The wiring diagram and simulation of HC-SR04	31
13	The code logic of HC-SR04 and SSD 1306 OLED	32
14	The simulation results of errors against the number of connecting sheets .	33
15	The entire workspace with different random indices	34
16	The entire workspace with different random indices	35
17	The cubic workspace in X-Z plane with H=300mm	35
18	The distance between random points and the nearest points on the edge with H=300mm	36
19	The top-down view of the cubic workspace	36
20	The cubic workspace with index=1000000, H=280mm	38
21	The kinematics model of manipulator with respective bending modules .	40
22	The cross-shaped trajectory and its replication by FABRIKc algorithm .	42
23	The closed trajectory and its replication by FABRIKc algorithm	42

List of Tables

1	The Characteristics of Different Manipulators.	3
2	The Parameters of Manipulators.	9
3	The Parameters of Manipulators.	37
4	The Parameters of Manipulators.	38
A.1	The results of FABRIKc algorithm with target angle $\alpha = [0\ 0\ 90\ 0]$	ix
A.2	The Singular Posture Solution by FABRIKc start with initial posture.	xii

1 Introduction

1.1 Background

This is the background part.

1.2 Motivation

This is the motivation part.

1.3 Introduction

Continuum robots has emerged and attracted a lot of attention since 2008 [1]. Before that, rigid joint robots were dominating the robotic arm industry. Compared with traditional rigid joint manipulator, continuum robots stand out for their flexible, highly bendable structure and extremely flexible motion performance. The limitations of rigid joint robots have gradually shown up in applications requiring highly detailed operation and in complex or space-limited environments. Thus, the continuum robot was developed and perfected during the years. This new kind of robot not only changes the code of traditional robot design but also demonstrates unprecedented application potential in fields such as exploration industry and medical science [2]. Meanwhile, rigid-flexible-soft coupled continuum robots combine the multiple advantages of the stability of rigid structures, the flexibility of bendable structures, and the compliance of soft structures, and are one of the most promising robots for increasingly complex tasks [3].

With unique bionic structure and motion characteristics, continuum robots provide new possibilities to solve these challenges.

This paper will discuss different types of existing continuum robots and their working principles, advantages, and disadvantages, then propose a proper continuum robot design that can be mainly applied to medical applications.

2 Literature Review

In the last two decades, significant progress in electronic and computer technologies has led to remarkable growth in the field of manipulator robotics. Manipulators developed by various institutions have been integrated into the industrial sector to autonomously or semi-autonomously perform repetitive tasks. Simultaneously, manipulators are utilized for tasks with stringent precision requirements to minimize errors. Additionally, essential tasks are undertaken by manipulators to substitute for humans in challenging environments.

2.1 Manipulators in Biology Field

With a growing emphasis on the field of biology, manipulators have also been introduced to provide assistance. In the field of medicine, manipulators have been utilized since the end of the last century. The AESOP robotic surgical system, proposed by Computer Motion founded by Yulun Wang in 1993, was investigated in laparoscopic surgery in 1997 [4]. Afterward, the ZEUS robotic surgical system endowed with a trilateral manipulator configuration was proposed by Computer Motion in 1998 [5]. During the period from 1999 to 2001, the ZEUS system was utilized for a series of clinical surgeries, demonstrating excellent performance [6–8]. At the beginning of the 21st century, a novel robotic system, da Vinci robotic surgery system, was designed to facilitate more intricate surgical procedures [9]. Moreover, manipulators can be leveraged in the field of biological physics as a viable approach for biological experiments. In the HIFU system developed by An et al., the SCARA (self compliant automatic robot assembly) robot was employed as manipulator, incorporating an ultrasound probe for the purpose of scanning biological tissues [10]. The robotic system FUSBOTs (Focal Ultrasound Surgery RoBOTs) was proposed and upgraded to accomplish more precise targeted treatment with multiple DoF (Degrees of Freedom) manipulators [11–13]. Despite the various utilization of manipulator platforms designed for accommodating ultrasonic transducers [14–16], certain limitations persist. Hence, a comparison of different manipulators is necessary in selection of appropriate type manipulator for integrating ultrasonic transducer.

2.2 Types of Manipulator

From the perspective of geometry, rigid-body manipulators can be categorized into two main types: parallel mechanisms and serial link [17]. The control of parallel mechanisms is relatively complex. Meanwhile, serial link manipulators can be further divided into five types: Cartesian (PPP), articulated (RRR), cylindrical (RPP), spherical (RRP), and SCARA (RRP). Additionally, with the advancement of robotics technology, two other types of manipulators, namely biomimetic and anthropomorphic, have demonstrated their advantages [18, 19]. The comparative analysis will be conducted to highlight the distinctive features of different manipulators presented in Table 1, leading to the identification of the most suitable type for specific applications.

Table 1: The Characteristics of Different Manipulators.

Manipulators	Types	DoF	Features	Applications
Stewart Platform [20]	Series-Parallel	6	series-parallel duality, mature kinematics algorithm.	flight simulation [21], robocrane [22]
Cartesian	PPP	3	technological maturity, low complexity, cost-effective, high payload.	3D-Printing framework [23], warehousing & hoisting [24], agricultural machinery [25]
Articulated [26]	RRR	6	factory automation, programmable, high precision and payload.	milling operations [27], material handling [28]
Cylindrical [29]	RPP	3	good precision, limited workspace.	assembling industries [30]
Spherical [31]	RRP	3	collision avoidance, low precision, long reach,	low-precision semi-automated tasks [27, 28]
SCARA [29]	RRP	3	light weight and fast operation.	surgical applications [32], medical rehabilitation [33]
Continuum/Soft [34, 35]	Biomimetic	∞	ability to handle fatigue objects, environmental adaptability.	tendon driven [36–38], fibre-reinforced [39, 40], fluid-elastic drive [41–43], bionics materials [44–46]

Continued on next page

Table 1 – continued from previous page

Manipulators	Types	DoF	Features	Applications
Humanoid	anthropomorphic	7	high efficiency, flexibility.	humanoid robots driven by pneumatic artificial muscles [47]

The distinctive structure of continuum manipulator imparts to the system with theoretically infinite DoF. The characteristic enables the continuum manipulator to navigate through confined spaces or manipulate objects along specific trajectories within the workspace [2, 34, 35]. Simultaneously, the continuum manipulator demonstrates suboptimal performance in the context of payload, making it challenging to meet the payload capacities commonly observed in industrial robotic systems. Nevertheless, within the realm of biology applications, the emphasis shifts towards precision and operability for manipulator. In this report, the manipulator platform integrated with the ultrasonic transducer module necessitates a design. The selection of the continuum manipulator was predicated upon its outstanding flexibility and the lightweight attributes of ultrasonic transducer module. The forthcoming discussion will delve into the categorisation of continuum robots, ultimately selecting an appropriate type to serve as the centerpiece for the manipulator platform.

2.3 Types of Continuum Robots

Nowadays, a variety of continuum robots exist, each exhibiting unique structures and functions. They serve in different fields such as medicine, construction, and exploration. In this section, some of the most popular continuum robots will be introduced, providing basic insights into their structures as well as discussing their merits and drawbacks.

2.3.1 Tendon-Driven Robots

The arm of the Tendon-Driven robot consists of a backbone, several tendons and disks. The backbone defines the structure and posture of the entire robot arm, while disks

define the diameter and divide the robot arm into segments, and tendons are stretched to create deformation and movements of different directions for the robot arm.

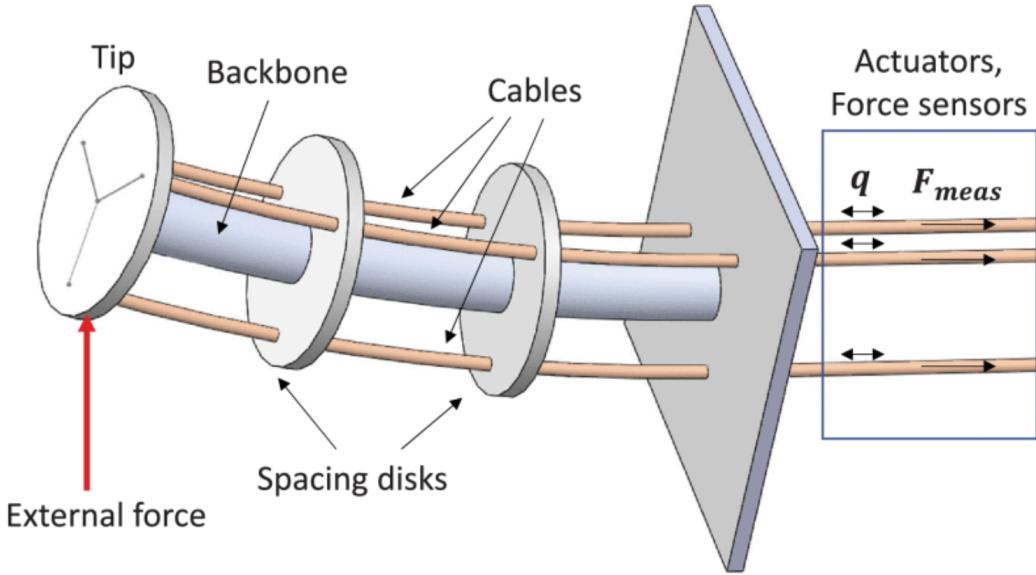


Figure 1: The three tendons continuum robot with one segment [48].

Figure 1 shows only the simplest tendon-driven robots. In practical applications, there may be more than one backbone, and the disks are not necessarily parallel.

Compared with other continuum robots, one of the most significant advantages of tendon-driven robots is their flexibility. This advantage makes it more effective in performing tasks in complex and restrictive spaces. In addition, due to the simple components needed to construct the robot, it is easier to meet lightweight design specifications. Moreover, like the concentric-tube continuum robots, tendon-driven continuum robots can be built designed on a small scale with diameters of below 10mm [49].

However, due to its simple actuating principle, more complex algorithms are needed to control it more accurately. Also, the tendon-driven robots actuate by pulling the tendon, which makes the friction between the tendon and other components inevitable, which will accelerate the wearing speed of the tendon-driven robots.

2.3.2 Fishbone Robots

Fishbone robots, as the name suggests, are inspired by fish bones. It consists of several "fishbone modules", which are composed of a number of rigid cross-shaped plates, and a flexible elastic plate backbone embedded in the middle, forming a rigid-soft coupling

structure[3]. When the different modules are connected, the backbones of the bionic fishbone modules are perpendicular to each other, and finally form a complete main frame of the fishbone robot. Like the tendon-driven robot, the fishbone robot is also controlled by cable, but it is worth noting that each module is controlled by two separate cables. This means that for a fishbone robot composed of n fishbone modules, there are a total of $2n$ cables controlling the motion of the whole robot. In addition, the planes formed by the two ropes that control each module are perpendicular to each other, so that each section rotates along a plane perpendicular to each other.

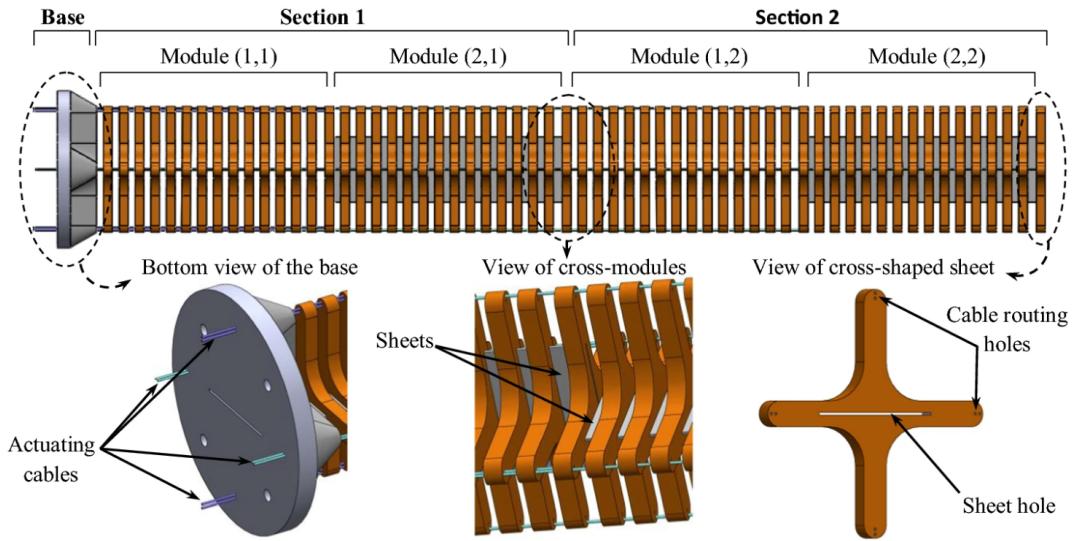


Figure 2: The cable-driven fish bone continuum robot with cable arrangement [50].

From the graph above, it is shown that because of the numbers of rigid cross-shaped sheets formed a large diameter frame, the structural stability of this kind of robot is very strong. Also, each fishbone unit can provide one DoF, making it easy to reach a very high total DoF. However, due to the multiple-curves deformation trajectory, the fishbone robots are not suitable for situations where strict trajectory is required[3].

2.3.3 Concentric Tube Continuum Robots

concentric tube robots, shaped like retractable walking sticks, consist of many tubes with decreasing diameters. Each tube is nested on top of the previous wider tube.

The concentric robots are made of two parts: tubes and coaxial actuation units. The tubes are the main structural element of this robot and act as the backbone. The coaxial

actuation unit consists of two motors which are responsible for rotation and translation movement respectively. Each tube is actuated by an independent coaxial actuation unit.

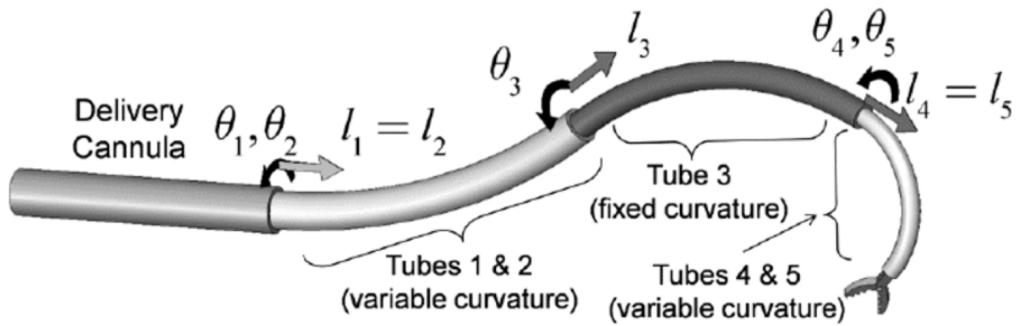


Figure 3: An example concentric tube continuum robot [51].

The most significant advantage of this kind of robots is that of all the continuum robots, concentric tube robots have the smallest possible outer diameter and are best suited to work in confined and narrow Spaces. Therefore, it is the ideal choice for surgical operations.

Their disadvantages, on the other hand, are also very evident. Since each tube requires an independent actuation unit, the overall length of the robot cannot be very long, because longer lengths will lead to more tubes, and will lead to more tip position errors.

3 Design

3.1 Structural Design

The novel Continuous Six-degree-of-freedom Manipulator introduced in this paper, which corresponds to the scheme design shown in Figure 2, belongs to the class of bio-inspired continuum robots [3]. In the design process, a nylon cable drive mechanism is employed as a substitute for the complex musculature of fish, enabling it to maintain stability while undergoing deformation movements, thus achieving effective deformation motion and controllability. As depicted in Figure 4, the rigid cross-shaped sheets is utilized to mimic the spine and ribs, uniformly arranging these sheets on a rubber pure soft sleeve to replace the spinal joints.

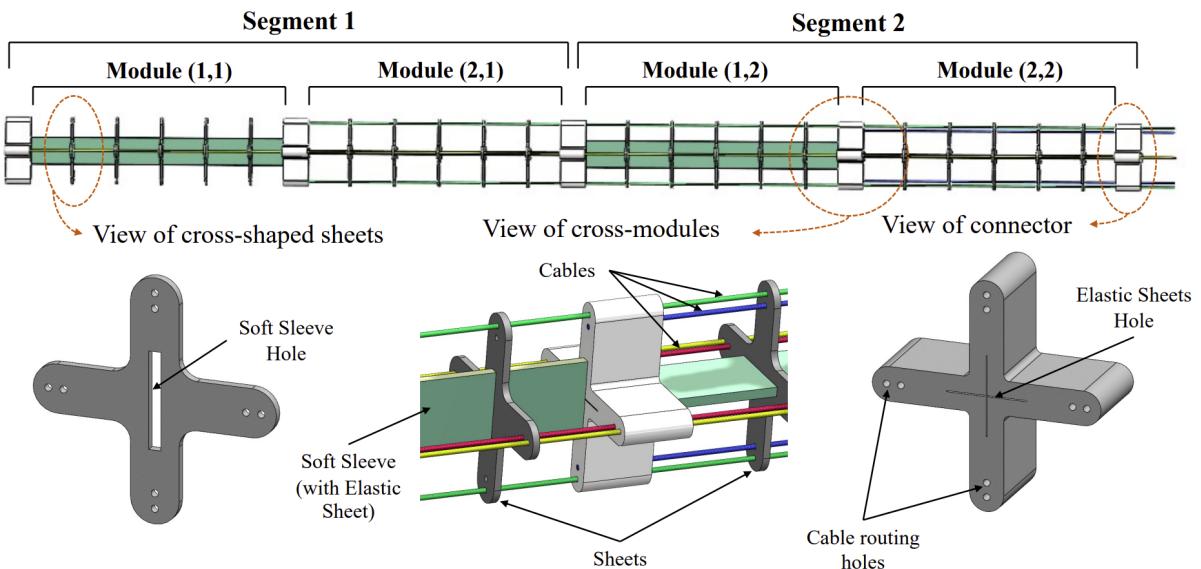


Figure 4: The main components of the proposed manipulator and cables arrangement.

3.2 Designation and Parameter Definition of Manipulator

The manipulator is composed of four identical modules, each of which can be independently controlled to bend within a two-dimensional space. Each module consists of an elastic sheet and five uniformly arranged cross-shaped sheets, where the rubber pure soft sleeve outside the elastic sheet is interference-fitted with the cross-shaped sheets. The connector end of each module features two symmetrically arranged inextensible cables that pass through the cross-shaped sheets and are anchored to the

connector, driving the end of each manipulator module. By applying tension to one of these cables, the elastic sheet can be deflected, thereby achieving a single degree of freedom planar motion at the end of the manipulator module. Since the elastic sheets of the modules are transversely mounted, the deflection of two or more sheets simultaneously allows for spatial motion at the end of the manipulator, as shown in Figure 4. Overall, the proposed manipulator comprises two segments, each maintaining a constant length of 675mm and a weight of 88 grams. Each module contains five cross-shaped sheets with a circumferential radius of 20mm and a thickness of 1.5mm. Both the head and the tail of each segment are connected by connectors, which have the same circumferential radius as the cross-shaped sheet and a thickness of 15mm.

The definition of manipulator parameters is presented in Table 2. The subsequent sections will adhere to the corresponding naming conventions.

Table 2: The Parameters of Manipulators.

Paramter	Definition	Value (mm)
Sr_i	the length of elastic sheet in Module i	$Sr_{1,2,3,4} = 150$
d_i	the thickness of the cross-shaped connector	$d_{1,2,3,4,5} = 15$
N	the number of cross-shaped sheet in a module	$N = 0 \sim 15$
r_i	the distance between the centroid of connector and cable routing hole	$r_{1,2} = 17.5$ $r_{3,4} = 15$
ΔS_i	the change rate of cable, cable _{2<i>i</i>-1} and cable _{2<i>i</i>} corresponds to Module i	

3.3 Material Selection

The fabrication of the Continuous Six-Degree-of-Freedom Manipulator primarily employs the following methods: Shape Deposition Manufacturing (SDM)[52], Casting[53, 54], and Multi-material 3D Printing Technologies[55, 56]. By integrating SDM with Casting techniques, it is possible to utilize rigid, flexible, and soft materials, and integrate components such as sensors and circuits into the structure, which is particularly crucial for the integrated manufacturing of sensors and actuators. On the

other hand, Multi-material 3D Printing technology supports the unified manufacturing of soft and hard materials, enabling the creation of structures with more complex geometries and adjustable material hardness according to specific requirements. This approach results in structures that are not only more scientifically arranged and exhibit a more rational combination of rigidity and flexibility but also possess higher stability and superior performance. Moreover, Multi-material 3D Printing enhances manufacturing efficiency. Given these advantages, 3D Printing technology emerges as the preferred method for constructing continuum robots.

The materials for the cross-shaped sheets and connectors can be fabricated in an integrated manner using Multi-material 3D Printers, which avoids the impact of imperfect assembly on deformation and achieves a compact design. The material chosen simulates engineering plastics (Mainly digital material with a Poisson's ratio of 0.394, a modulus of elasticity of 2.2 GPa and a Young's modulus of 2.5 GPa). The soft sleeve material is made of silicone rubber, characterized by a Poisson's ratio of 0.47. Furthermore, a 65Mn alloy steel (with a Poisson's ratio of 0.244 and a Young's modulus of 0.2 GPa) elastic sheet (flexible) embedded within the soft sleeve forms a bio-inspired sheath, creating a bio-inspired fishbone structural unit. This structural unit is a hybrid of rigidity, flexibility, and softness, not only exhibiting excellent constant curvature characteristics but also higher structural stability. The bio-inspired fishbone unit also provides a constraint mechanism for the synchronous bending of the driving cables, effectively reducing the impact of external mechanical collisions on the cables [3]. Therefore, this unique structural design is a key foundation for achieving high-precision modelling of continuum robots.

In this design, the elastic sheet is combined with the connector by embedding it into a 0.3mm wide slot on the connector and utilizing an interference fit, a method that eliminates the need for additional fasteners and ensures the structural integrity of the continuum robot during standard operations. Nonetheless, to prevent the robot from separating under extreme overload conditions, special fixing holes were designed on both the elastic sheet and the connector. The installation of screws through these holes enhances the robustness of the continuum robot in harsh environments. Leveraging the modular design characteristic of the continuum robot, it can be constructed by serially connecting multiple similar motion modules. The example demonstrated in this study is

comprised of two such motion modules, as shown in Figure 4, whereby altering the length of the driving cables enables the continuum robot to achieve multi-degree of freedom movements.

Figure 4 illustrates the cable configuration scheme proposed for the continuum robot in this paper. Each bio-inspired fishbone structural unit is actuated by two symmetrically arranged cables, which traverse through the guide holes of the connectors and cross-shaped sheets, with each module being controlled by two motors, totaling eight motors. The guide holes in the cross-shaped sheets ensure that the cables maintain an arc shape within their bending region. In the actual design process, the maximum number of cross-shaped sheets is determined by the maximum bending angle preset for the bio-inspired fishbone unit. Through simulation analysis, we have concluded that the bending performance of the elastic sheet is optimal when the bending angle reaches 73°. The design proposal offers the following advantages:

- Lightweight: Utilizing cross-shaped sheets results in a lighter weight compared to traditional cylindrical manipulators.
- Stable Deformation Motion: The deformation motion is stable, offering an improvement over traditional cylindrical manipulator.
- Rigidity: The adoption of a stacked skeleton structure effectively eliminates bending deformation.
- Energy Consumption [57]: The proposed robot design facilitates a reduction in the energy required to achieve spatial movement; that is, each module employs two actuation motors instead of three, as is the case with continuum robots possessing a cylindrical backbone.

Like all continuum robots, the sole drawback of this design is the complexity involved in modeling.

3.4 Methodology

3.4.1 Forward Kinematics

To derive the workspace of the manipulator for further analysis, the forward kinematics algorithm of the manipulator need to be conducted. The forward kinematics algorithm of the fishbone continuum robot[3] with one segment is given. However, utilizing this algorithm for calculation becomes complex while there are four modules in the manipulator. Additionally, the inverse kinematics algorithm also requires the assistance of forward kinematics algorithm for subsequent calculations using the composite coordinate transformation formula. Therefore, The relevant forward kinematics algorithm for subsequent calculations need to be derived.

The base coordinate system can be established with the centroid of base connector upper surface serving as the origin. The x-axis of the coordinate system is parallel to the backbone of Module 1, which is Module (2,2). Consequently, the backbones of Modules 1 and 3, which are Modules (2,2) and (2,1) are parallel to the x-axis of base coordinate system, while the backbones of Modules 2 and 4, which are Modules (1,2) and (1,1) are parallel to the y-axis of base coordinate system. The positions of the five centroids in the base coordinate system when the manipulator is in the initial posture are shown in Figure 5. The upper surface centroids of the connectors are designated as $node_1$, $node_2$, $node_3$, $node_4$, and $node_5$.

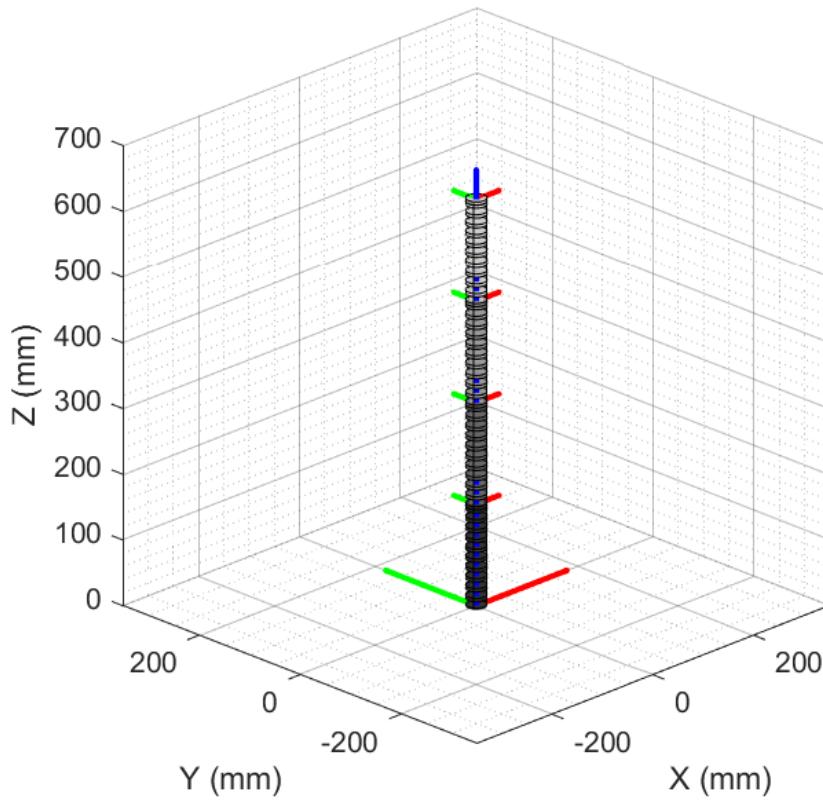


Figure 5: The kinematics model of manipulator in initial position.

The Module 1 is restricted to bending in the y-z plane of the coordinate system where $node_1$ serves as the origin, while the Module 2 is restricted to bending in the x-z plane of the coordinate system where $node_2$ serves as the origin. Similarly, the Modules 3 and 4 are subject to the same constraints. The bending angles for these modules are defined as α_1 , α_2 , α_3 , and α_4 , respectively. The positions of the manipulator model in the base coordinate system after bending each module by 90° are illustrated in Figure 6.

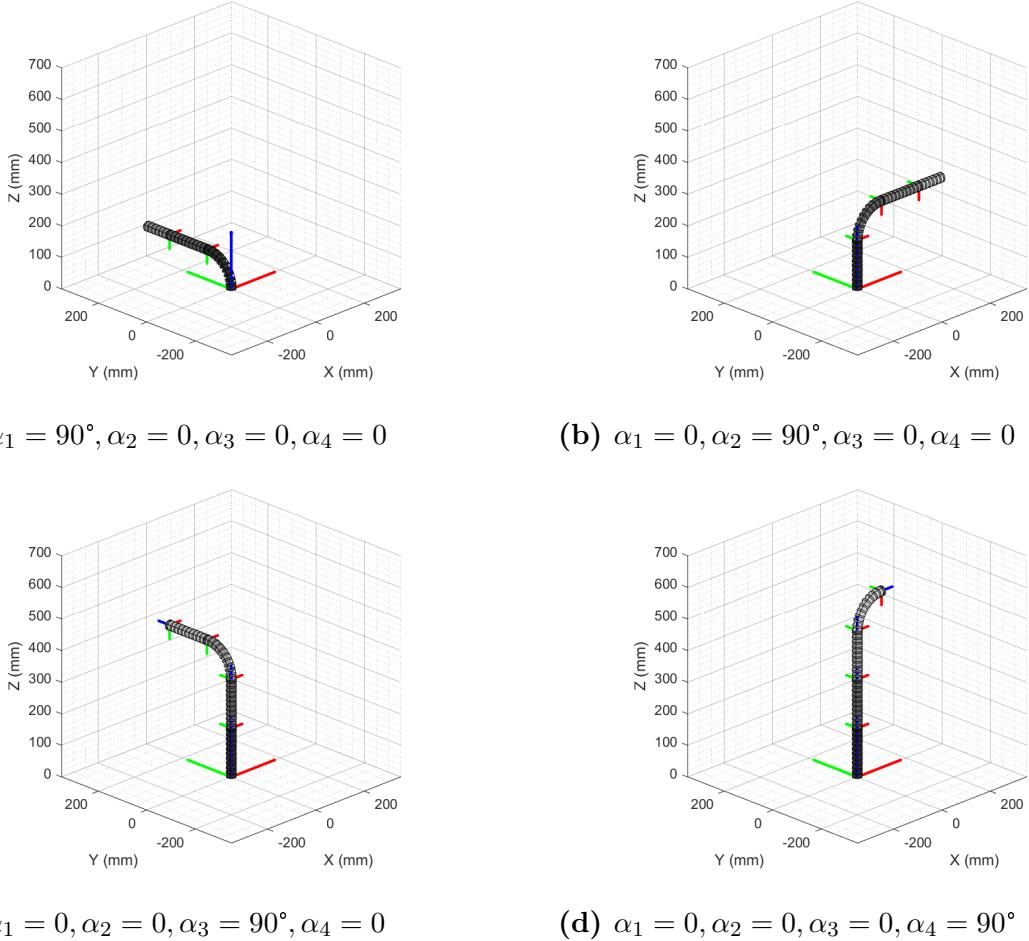


Figure 6: The kinematics model of manipulator with respective bending modules.

While the modules bend to the positive direction of x-axis or y-axis, the bending angles α are positive. Owing to the distinct properties of the four modules, different calculation methods are required for analysis. The Module i have a base node $node_i$ and an end effector node $node_{i+1}$. To further calculate the position of $node_{i+1}$ in the base coordinate system, these matrices can be employed in the Equation 1.

$$\mathbf{P}_{i+1}^{base} = \mathbf{R}_i \times \mathbf{P}_{i+1}^i + \mathbf{P}_i^{base} \quad (1)$$

\mathbf{P}_{i+1}^{base} : The position of $node_{i+1}$ in the base coordinate system.

\mathbf{R}_i : The rotational matrix transforms the base coordinate system into coordinate system i , which is the coordinate system with origin $node_i$.

\mathbf{P}_{i+1}^i : The position of $node_{i+1}$ in coordinate system i .

\mathbf{P}_i^{base} : The position of $node_i$ in the base coordinate system.

- Module 1

For Module 1, the relative position matrix of $node_1$ and $node_2$ \mathbf{P}_2^1 and α_1 is shown in Equation 2. R_1 is the radius of the Module 1 bending curve. The rotational matrix \mathbf{R}_1 and the position matrix of $node_1$ \mathbf{P}_1^{base} are shown in Equations 3 and 4.

$$\mathbf{P}_2^1 = \begin{bmatrix} 0 \\ (R_1 \cdot (1 - \cos(\alpha_1)) + d_2 \cdot \sin(\alpha_1)) \\ (R_1 \cdot \sin(\alpha_1) + d_2 \cdot \cos(\alpha_1)) \end{bmatrix} \quad (2)$$

(hint : $R_1 = Sr_1/\alpha_1$)

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$\mathbf{P}_1^{base} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \quad (4)$$

According to the Equations 2, 3, and 4, the position matrix of $node_2$ in the base coordinate system \mathbf{P}_2^{base} can be calculated in Equation 5.

$$\begin{aligned} \mathbf{P}_2^{base} &= \mathbf{R}_1 \times \mathbf{P}_2^1 + \mathbf{P}_1^{base} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ (R_1 \cdot (1 - \cos(\alpha_1)) + d_2 \cdot \sin(\alpha_1)) \\ (R_1 \cdot \sin(\alpha_1) + d_2 \cdot \cos(\alpha_1)) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ (R_1 \cdot (1 - \cos(\alpha_1)) + d_2 \cdot \sin(\alpha_1)) \\ (R_1 \cdot \sin(\alpha_1) + d_2 \cdot \cos(\alpha_1)) \end{bmatrix} \end{aligned} \quad (5)$$

- Module 2

For Module 2, the relationship between the relative position matrix of $node_2$ and $node_3$ \mathbf{P}_3^2 and α_2 is shown in Equation 6. R_2 is the radius of the Module 2 bending curve. The rotational matrix \mathbf{R}_2 and the position matrix of $node_2$ \mathbf{P}_2^{base} are shown

in Equations 7 and 5.

$$\mathbf{P}_3^2 = \begin{bmatrix} (R_2 \cdot (1 - \cos(\alpha_2)) + d_3 \cdot \sin(\alpha_2)) \\ 0 \\ (R_2 \cdot \sin(\alpha_2) + d_3 \cdot \cos(\alpha_2)) \end{bmatrix} \quad (6)$$

(hint : $R_2 = S r_2 / \alpha_2$)

$$\mathbf{R}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1) & \sin(\alpha_1) \\ 0 & -\sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \quad (7)$$

According to the Equations 5, 6, and 7, the position matrix of *node*₃ in the base coordinate system \mathbf{P}_3^{base} can be calculated in Equation 8.

$$\begin{aligned} \mathbf{P}_3^{base} &= \mathbf{R}_1 \times \mathbf{R}_2 \times \mathbf{P}_3^2 + \mathbf{P}_2^{base} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1) & \sin(\alpha_1) \\ 0 & -\sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \\ &\quad \times \begin{bmatrix} (R_2 \cdot (1 - \cos(\alpha_2)) + d_3 \cdot \sin(\alpha_2)) \\ 0 \\ (R_2 \cdot \sin(\alpha_2) + d_3 \cdot \cos(\alpha_2)) \end{bmatrix} \\ &\quad + \begin{bmatrix} (R_1 \cdot (1 - \cos(\alpha_1)) + d_2 \cdot \sin(\alpha_1)) \\ 0 \\ (R_1 \cdot \sin(\alpha_1) + d_2 \cdot \cos(\alpha_1)) \end{bmatrix} \end{aligned} \quad (8)$$

- Module 3

For Module 3, the relationship between the relative position matrix of *node*₃ and *node*₄ \mathbf{P}_4^3 and α_3 is shown in Equation 9. R_3 is the radius of the Module 3 bending curve. The rotational matrix \mathbf{R}_3 and the position matrix of *node*₃ \mathbf{P}_3^{base} are shown

in Equations 10 and 8.

$$\mathbf{P}_4^3 = \begin{bmatrix} 0 \\ (R_3 \cdot (1 - \cos(\alpha_3)) + d_4 \cdot \sin(\alpha_3)) \\ (R_3 \cdot \sin(\alpha_3) + d_4 \cdot \cos(\alpha_3)) \end{bmatrix} \quad (9)$$

(hint : $R_2 = Sr_2/\alpha_2$)

$$\mathbf{R}_3 = \begin{bmatrix} \cos(\alpha_2) & 0 & \sin(\alpha_2) \\ 0 & 1 & 0 \\ -\sin(\alpha_2) & 0 & \cos(\alpha_2) \end{bmatrix} \quad (10)$$

According to the Equations 8, 9, and 10, the position matrix of *node*₄ in the base coordinate system \mathbf{P}_4^{base} can be calculated in Equation 11.

$$\begin{aligned} \mathbf{P}_4^{base} &= \mathbf{R}_1 \times \mathbf{R}_2 \times \mathbf{R}_3 \times \mathbf{P}_4^3 + \mathbf{P}_3^{base} \\ &= \mathbf{R}_1 \times \mathbf{R}_2 \times \begin{bmatrix} \cos(\alpha_2) & 0 & \sin(\alpha_2) \\ 0 & 1 & 0 \\ -\sin(\alpha_2) & 0 & \cos(\alpha_2) \end{bmatrix} \\ &\times \begin{bmatrix} 0 \\ (R_3 \cdot (1 - \cos(\alpha_3)) + d_4 \cdot \sin(\alpha_3)) \\ (R_3 \cdot \sin(\alpha_3) + d_4 \cdot \cos(\alpha_3)) \end{bmatrix} + \mathbf{P}_3^{base} \end{aligned} \quad (11)$$

- Module 4

In summary, the position matrix of *node*₅ in the base coordinate system P_5^{base} can be represented by Equation 12.

$$\mathbf{P}_5^{base} = \mathbf{R}_1 \times \mathbf{R}_2 \times \mathbf{R}_3 \times \mathbf{R}_4 \times \mathbf{P}_4^4 + \mathbf{P}_4^{base} \quad (12)$$

Applying the corresponding transformations to the Equation 1 reveals the patterns shown in Equations 13 and 14.

$$\mathbf{P}_{i+1}^{base} - \mathbf{P}_i^{base} = \prod_{n=1}^i \mathbf{R}_n \times \mathbf{P}_{i+1}^i \quad (13)$$

$$\mathbf{P}_{i+1}^{base} = \sum_{m=1}^i \left[\prod_{n=1}^m \mathbf{R}_n \times \mathbf{P}_{m+1}^m \right] \quad (\mathbf{P}_1^{base} = 0) \quad (14)$$

To emphasize transformation of manipulator, homogeneous transformation matrices [58] are employed to describe the position of the end effector of the manipulator in Equations 15 and 16.

$$\mathbf{H}_{i+1}^i = \begin{bmatrix} \mathbf{R}_{i+1}^i & \mathbf{P}_{i+1}^i \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} \mathbf{P}_{i+1}^{base} \\ 1 \end{bmatrix} = \prod_{j=1}^i \mathbf{H}_{j+1}^j \times \begin{bmatrix} \mathbf{P}_{i+1}^i \\ 1 \end{bmatrix} \quad (16)$$

3.4.2 Inverse Kinematics

The inverse kinematics algorithm is inspired by a rigid manipulator inverse kinematics solver called FABRIK [59]. W. Zhang et al.[60] expanded this solver to the continuum manipulator domain. Considering the distinctive nature of the manipulator, adaptations have been implemented to the FABRIKc solver, transforming it into a tailored inverse kinematics algorithm for this particular application. For further elucidation, the inverse kinematics derivation is demonstrated using the posture where $\alpha_1 = 0$, $\alpha_2 = 0$, $\alpha_3 = 90$, and $\alpha_4 = 0$. The comprehensive process of the first epoch will be elaborated. The iterative procedure of epoch 1 is elucidated as follow. Simultaneously, the flowchart of FABRIKc algorithm is shown in Figure 7.

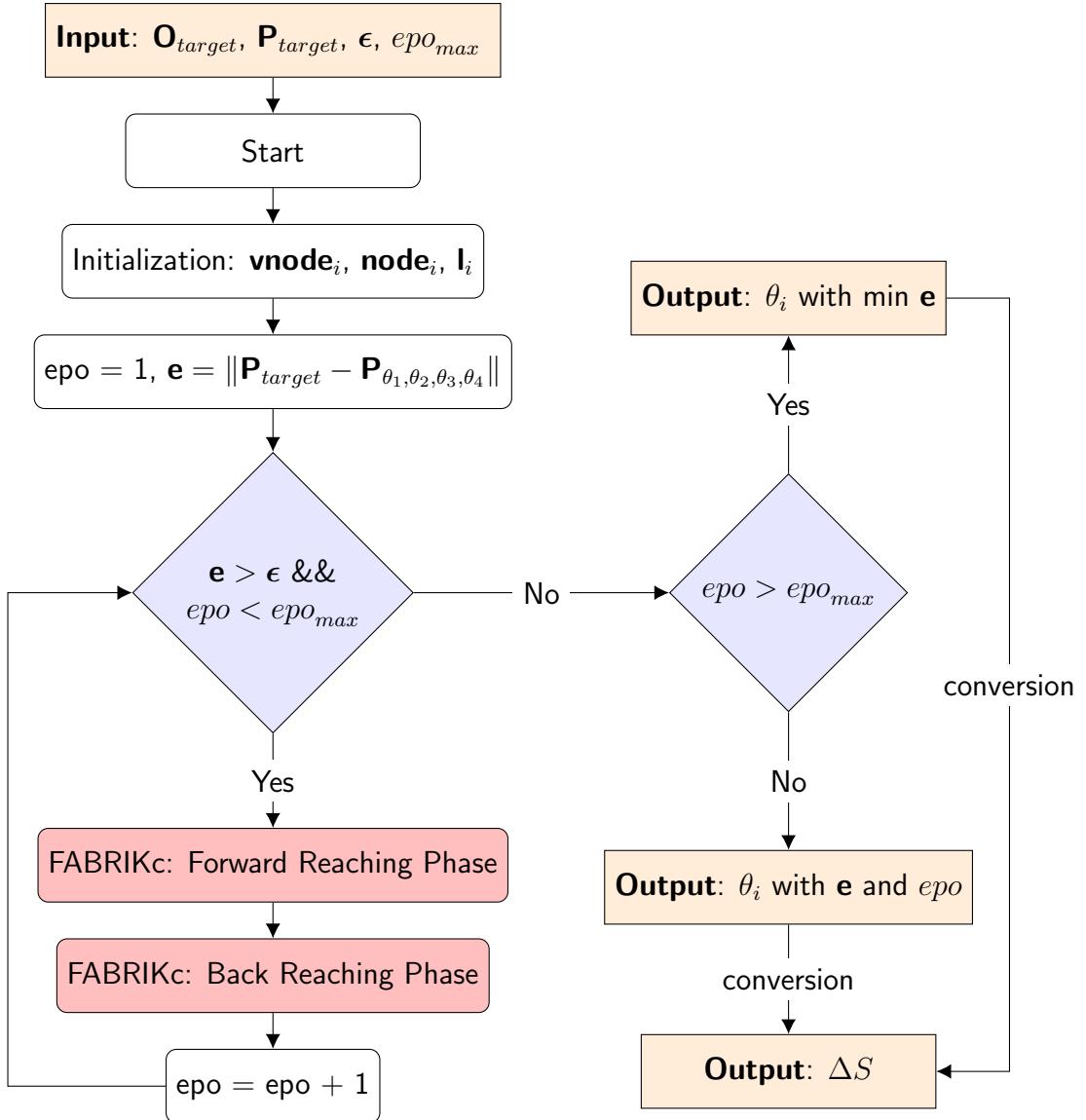


Figure 7: The flow chart of the FABRIKc algorithm.

Pimarily, the position and orientation of end effector can be calculated by forward kinematics algorithm conducted in Equation 15. The orientation and position matrices of the target are derived in Equation 17. These matrices are utilized as the input of the FABRIKc algorithm. The output of the algorithm is a series of bending angles $\theta_1, \theta_2, \theta_3$, and θ_4 .

$$\mathbf{O}_{target} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{P}_{target} = \begin{bmatrix} 0 \\ 245.493 \\ 395.493 \end{bmatrix} \quad (17)$$

Additionally, it is essential to introduce a novel concept called "virtual node". This arised

from the fact that FABRIK algorithm was originally designed for rigid manipulators, which implied that continuum manipulators need to be abstracted for modeling purpose. In this content, the module of manipulator are abstracted as a rigid manipulator with two prismatic joints and one revolute joint. The location of the revolute joint serves as the virtual node.

Subsequently, the forward reaching phase of the FABRIKc algorithm can commence. The calculation which involves four iterations is specified as follow.

- Initialization

The manipulator is estimated to stay at the initial position, which is shown in Figure 5. Under this circumstance, the nodes and virtual nodes of the manipulator **node**_i and **vnode**_i are listed in Equations 18 and 19. The connector thickness of manipulator are not taken into account in this demonstration, despite being considered in the programme. The virtual length **l**_i for Module i, which is the distance from **vnode**_i to **node**_i as well as *node*_{i+1}, is derived in Equation 20.

$$\begin{aligned} \mathbf{node}_1 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \mathbf{node}_2 &= \begin{bmatrix} 0 \\ 0 \\ 150 \end{bmatrix} & \mathbf{node}_3 &= \begin{bmatrix} 0 \\ 0 \\ 300 \end{bmatrix} & \mathbf{node}_4 &= \begin{bmatrix} 0 \\ 0 \\ 450 \end{bmatrix} & \mathbf{node}_5 &= \begin{bmatrix} 0 \\ 0 \\ 600 \end{bmatrix} \end{aligned} \quad (18)$$

$$\begin{aligned} \mathbf{vnode}_1 &= \begin{bmatrix} 0 \\ 0 \\ 75 \end{bmatrix} & \mathbf{vnode}_2 &= \begin{bmatrix} 0 \\ 0 \\ 225 \end{bmatrix} & \mathbf{vnode}_3 &= \begin{bmatrix} 0 \\ 0 \\ 375 \end{bmatrix} & \mathbf{vnode}_4 &= \begin{bmatrix} 0 \\ 0 \\ 525 \end{bmatrix} \end{aligned} \quad (19)$$

$$\mathbf{l}_i = \frac{Sr_i}{\theta_i} \cdot \tan(\theta_i) \quad (\text{hint : } \mathbf{l}_i = Sr_i/2 \text{ while } \theta_i = 0) \quad (20)$$

$$\mathbf{l}_1 = \mathbf{l}_2 = \mathbf{l}_3 = \mathbf{l}_4 = 75 \text{ (unit : mm)}$$

- Iteration 1

The virtual node of Module 4 can be derived by \mathbf{O}_{target} and \mathbf{P}_{target} in Equation 17. The virtual node **vnode**₄ can be derived by Equation 21.

$$\mathbf{vnode}'_4 = \mathbf{P}_{target} - \mathbf{l}_4 \cdot \mathbf{O}_{target}^z \quad (\text{hint : } z \text{ is } z\text{-axis orientation}) \quad (21)$$

Afterwards, the vector from virtual node of Module 3 to virtual node of Module 4 can be derived based on the positions of \mathbf{vnode}_3 and \mathbf{vnode}'_4 . The coordinate of the vector in the orientation of \mathbf{vnode}'_4 can be calculated in Equation 22.

$$\mathbf{vector}_4 = \mathbf{O}_{target} \times (\mathbf{vnode}'_4 - \mathbf{vnode}_3) \quad (22)$$

The bending angle of Module 4 θ_4 can be derived in Equation 23. The y-axis directional component is neglected because Module 4 can only bend in x-z plane. The updated length \mathbf{l}_4^{update} can be calculated according to Equation 24. Meanwhile, the updated $\mathbf{vnode}_4^{update}$ can be recalculated using \mathbf{l}_4^{update} in Equation 25. The \mathbf{node}_4 can also be determined based on the orientation of $\mathbf{vnode}_4^{update}$, which is \mathbf{O}_{vnode_4} , in Equation 27.

$$\theta_4 = -\arctan2(\mathbf{vector}_4^x, \mathbf{vector}_4^z) \quad (23)$$

$$\mathbf{l}_4^{update} = \frac{Sr_4}{\theta_4} \cdot \tan(\theta_4) \quad (24)$$

$$\mathbf{vnode}_4^{update} = \mathbf{P}_{target} - \mathbf{l}_4^{update} \cdot \mathbf{O}_{target}^z \quad (25)$$

$$\mathbf{O}_{vnode_4} = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) \\ 0 & 1 & 0 \\ -\sin(\theta_4) & 0 & \cos(\theta_4) \end{bmatrix} \times \mathbf{O}_{target} \quad (26)$$

$$\mathbf{node}_4 = \mathbf{vnode}_4^{update} - \mathbf{l}_4^{update} \cdot \mathbf{O}_{vnode_4}^z \quad (27)$$

- Iteration 2

The virtual node of Module 3 can be derived by \mathbf{O}_{vnode_4} and \mathbf{node}_4 in Equations 26 and 27. The virtual node \mathbf{vnode}_3 can be derived by Equation 28.

$$\mathbf{vnode}'_3 = \mathbf{node}_4 - \mathbf{l}_3 \cdot \mathbf{O}_{vnode_4}^z \quad (28)$$

Afterwards, the vector from virtual node of Module 2 to virtual node of Module 3 can be derived based on the positions of \mathbf{vnode}_2 and \mathbf{vnode}'_3 . The coordinate of the vector in the orientation of \mathbf{vnode}'_3 can be calculated in Equation 29.

$$\mathbf{vector}_3 = \mathbf{O}_{vnode_3} \times (\mathbf{vnode}'_3 - \mathbf{vnode}_2) \quad (29)$$

The bending angle of Module 3 θ_3 can be derived in Equation 30. The x-axis directional component is neglected because Module 3 can only bend in y-z plane. The updated length \mathbf{l}_3^{update} can be calculated according to Equation 31. Meanwhile, the updated $\mathbf{vnode}_3^{update}$ can be recalculated using \mathbf{l}_3^{update} in Equation 32. The \mathbf{node}_3 can also be determined based on the orientation of $\mathbf{vnode}_3^{update}$, which is \mathbf{O}_{vnode_3} , in Equation 34.

$$\theta_3 = -\arctan2(\mathbf{vector}_3^y, \mathbf{vector}_3^z) \quad (30)$$

$$\mathbf{l}_3^{update} = \frac{Sr_3}{\theta_3} \cdot \tan(\theta_3) \quad (31)$$

$$\mathbf{vnode}_3^{update} = \mathbf{node}_4 - \mathbf{l}_3^{update} \cdot \mathbf{O}_{vnode_4}^z \quad (32)$$

$$\mathbf{O}_{vnode_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_3) & \sin(\theta_3) \\ 0 & -\sin(\theta_3) & \sin(\theta_3) \end{bmatrix} \times \mathbf{O}_{vnode_4} \quad (33)$$

$$\mathbf{node}_3 = \mathbf{vnode}_3^{update} - \mathbf{l}_3^{update} \cdot \mathbf{O}_{vnode_3}^z \quad (34)$$

- Iteration 3

The virtual node of Module 2 can be derived by \mathbf{O}_{vnode_3} and \mathbf{node}_3 in Equations 33 and 34. The virtual node \mathbf{vnode}_2 can be derived by Equation 35.

$$\mathbf{vnode}'_2 = \mathbf{node}_3 - \mathbf{l}_2 \cdot \mathbf{O}_{vnode_3}^z \quad (35)$$

Afterwards, the vector from virtual node of Module 1 to virtual node of Module 2 can be derived based on the positions of \mathbf{vnode}_1 and \mathbf{vnode}'_2 . The coordinate of the vector in the orientation of \mathbf{vnode}'_2 can be calculated in Equation 36.

$$\mathbf{vector}_2 = \mathbf{O}_{vnode_2} \times (\mathbf{vnode}'_2 - \mathbf{vnode}_1) \quad (36)$$

The bending angle of Module 2 θ_2 can be derived in Equation 37. The y-axis directional component is neglected because Module 2 can only bend in x-z plane. The updated length \mathbf{l}_2^{update} can be calculated according to Equation 38. Meanwhile, the updated $\mathbf{vnode}_2^{update}$ can be recalculated using \mathbf{l}_2^{update} in Equation 39. The \mathbf{node}_2 can also be determined based on the orientation of $\mathbf{vnode}_2^{update}$, which is

\mathbf{O}_{vnode_2} , in Equation 41.

$$\theta_2 = -\arctan2(\mathbf{vector}_2^x, \mathbf{vector}_2^z) \quad (37)$$

$$\mathbf{l}_2^{update} = \frac{Sr_2}{\theta_2} \cdot \tan(\theta_2) \quad (38)$$

$$\mathbf{vnode}_2^{update} = \mathbf{node}_3 - \mathbf{l}_2^{update} \cdot \mathbf{O}_{vnode_3}^z \quad (39)$$

$$\mathbf{O}_{vnode_2} = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix} \times \mathbf{O}_{vnode_3} \quad (40)$$

$$\mathbf{node}_2 = \mathbf{vnode}_2^{update} - \mathbf{l}_2^{update} \cdot \mathbf{O}_{vnode_2}^z \quad (41)$$

- Iteration 4

The virtual node of Module 1 can be derived by \mathbf{O}_{vnode_2} and \mathbf{node}_2 in Equations 40 and 41. The virtual node \mathbf{vnode}_1 can be derived by Equation 42.

$$\mathbf{vnode}'_1 = \mathbf{node}_2 - \mathbf{l}_1 \cdot \mathbf{O}_{vnode_2}^z \quad (42)$$

Afterwards, the coordinate of the vector in the orientation of \mathbf{vnode}'_1 can be directly calculated in Equation 43. This is because the $\mathbf{O}_{node_1}^z$ must be oriented vertically upward.

$$\mathbf{vector}_2 = \mathbf{O}_{vnode_1} \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad (43)$$

The bending angle of Module 1 θ_1 can be derived in Equation 44. The x-axis directional component is neglected because Module 1 can only bend in y-z plane. The updated length \mathbf{l}_1^{update} can be calculated according to Equation 45. Meanwhile, the updated $\mathbf{vnode}_1^{update}$ can be recalculated using \mathbf{l}_1^{update} in Equation 46. The \mathbf{node}_1 can also be determined based on the orientation of $\mathbf{vnode}_1^{update}$, which is \mathbf{O}_{vnode_1} , in Equation 48.

$$\theta_1 = -\arctan2(\mathbf{vector}_1^y, \mathbf{vector}_1^z) \quad (44)$$

$$\mathbf{l}_1^{update} = \frac{Sr_1}{\theta_1} \cdot \tan(\theta_1) \quad (45)$$

$$\mathbf{vnode}_1^{update} = \mathbf{node}_2 - \mathbf{l}_1^{update} \cdot \mathbf{O}_{vnode_2}^z \quad (46)$$

$$\mathbf{O}_{vnode_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_1) \\ 0 & -\sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \times \mathbf{O}_{vnode_2} \quad (47)$$

$$\mathbf{node}_1 = \mathbf{vnode}_1^{update} - \mathbf{l}_1^{update} \cdot \mathbf{O}_{vnode_1}^z \quad (48)$$

Ultimately, the FABRIKc algorithm initiates the backward reaching phase, which is forward kinematics algorithm. The FABRIKc algorithm inherently guarantees the orientation of manipulator end effector, with the necessity to account solely for positional errors. The error can be calculated by Equation 49.

$$\mathbf{e} = \|\mathbf{P}_{target} - \mathbf{P}_{\theta_1, \theta_2, \theta_3, \theta_4}\| \quad (49)$$

For the first epoch, the results of FABRIKc algorithm are $\boldsymbol{\theta} = [6.71, -0.0, 83.29, -0.0]^\circ$, The error of the first epoch is 33.77506. The results about FABRIKc algorithm with the target angle $\boldsymbol{\alpha} = [0, 0, 90, 0]$ are shown in Table A.1 in Appendix A. The algorithm has been configured to execute a maximum of 200 epochs, with the provision to pause when the error reaches a sufficiently low level, thereby conserving computational resources. The maximum epoch number is defined as $epoch_{max}$. The error margin is ϵ .

3.5 Electronic Control

3.5.1 Manipulator Actuation Control

As mentioned above, the manipulator is divided into four units in total, with each unit being controlled for its bending movements by two cables. This means that the entire manipulator is controlled by a total of 8 cables. How to control the extension and retraction of these 8 cable ropes is the key to the Control part.

In this project, 8 stepper motors are employed to control the 8 individual cables. The planned design involves wrapping the cables around the rotor of each motor, and the stepper motors are programmed to extend or retract the ropes by stepping a certain number of steps in either a counterclockwise or clockwise direction. These 8 motors are connected to an Arduino board together, and the user can control the program by a keyboard, and a serial monitor.

- Actuation Principle

In this program, the user inputs the desired length changes for 8 cables $\Delta S_1 \dots \Delta S_8$ ($\Delta S_{1,2,3,4}$ ranging from -36.65mm to 36.65mm, $\Delta S_{5,6,7,8}$ ranging from -68.07 to 68.07mm). After conversion, it outputs the corresponding step numbers $Step_1 \dots Step_8$ for the eight stepper motors. Based on this, the 8-stepper motors will step the correspondingly. The relationship between the changes in cable length and the motor steps can be defined by Equation 50.

$$Step_i = \frac{(\Delta S_i - \Delta S_{i(prev)}) \cdot N}{\pi \cdot d}, \quad i \in [1, 8] \quad (50)$$

$\Delta S_{i(prev)}$: The length change for i^{th} cable in the previous location.

d : The diameter of the rotor. A rotor of diameter 10mm is used currently.

N : Number of steps/rev. 2048 for 28YBG-48 motor.

- Component Selection

To achieve a successful and precise control functionality, the first thing to do is to select appropriate components. Below are the main electronic components chosen:

- Arduino Mega 2560 board:

Because the project requires connecting 8 motors at the same time, the board should have more pins on the board as well as stronger processing capabilities compared to ordinary boards. Therefore, the Arduino Mega 2560 has been selected. It boasts up to 54 I/O pins and a more powerful processor to allow for better performance in multitasking operations [61].

- 28YBJ-48 stepper motor:

The 28YBJ-48 stepper motor is currently one of the most commonly used stepper motors on the market. It operates in two modes: half-step and full-step, corresponding to 4096 steps/rev and 2048 steps/rev respectively. This high-resolution configuration allows the manipulator to perform high-precision operations. Additionally, at its rated voltage, it can provide a torque of up to 34Nm, which is more than sufficient for pulling the cables of a lightweight

mechanical arm.

- ULN 2003A motor control chip:

The motor control chip acts as the brain of the motor, responsible for translating instructions from the Arduino board into operations that the motor can execute. The ULN2003A chip is specifically matched with the 28YBJ-48 motor, ensuring seamless coordination and operation between the chip and the motor.

- 9V power supply:

The rated operating voltage of the stepper motor is 5-12V. Here, a 9V power supply is chosen to allow losses due to component resistance.

- Circuit Schematics

The schematic of the overall circuit for the manipulator actuation control part is shown in Figure 8.

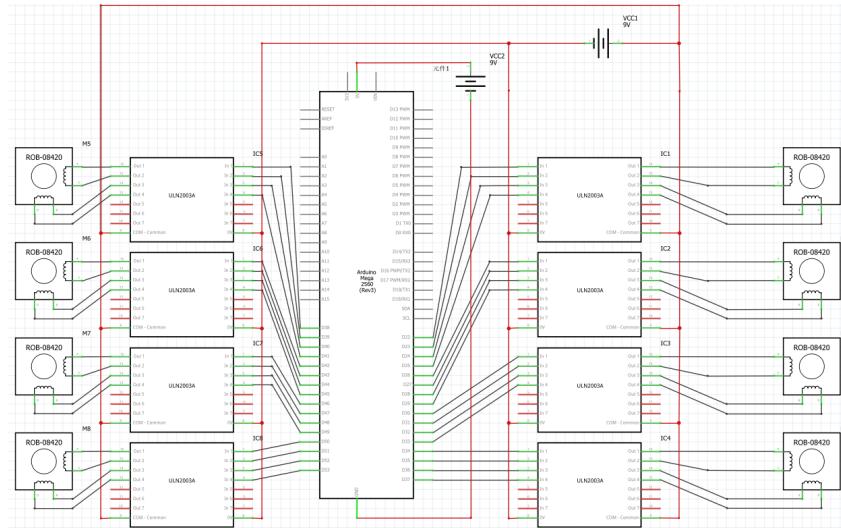


Figure 8: The circuit layout of Arduino step motor control system.

The 8 motors use pin 23-52 on the Arduino Mega 2560 board, and are powered by a 9V battery. The power supply is in a parallel configuration on a breadboard. The Arduino board is connected to a serial monitor and a keyboard for state inspection and manual input.

- Program development

The main logic framework of the program has been illustrated by the flowchart shown in Figure 9.

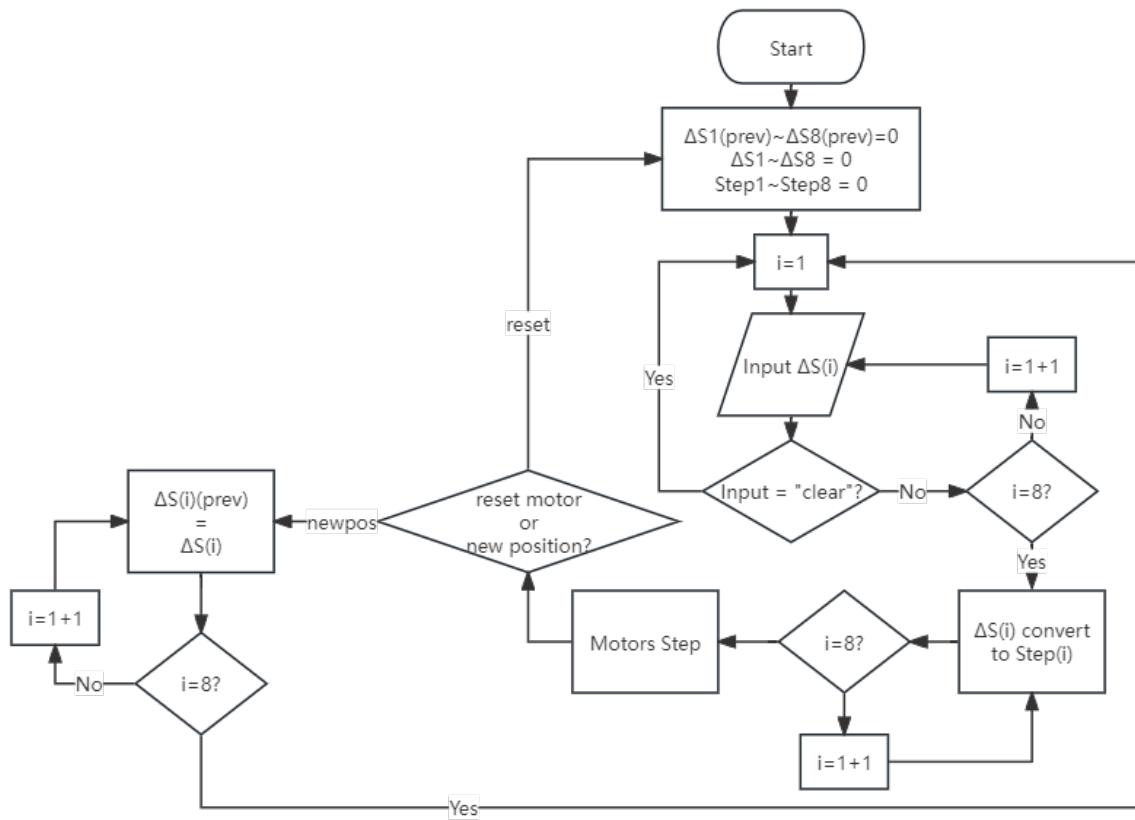


Figure 9: The flow chart of Arduino step motor control programme.

After the program starts, the first thing is to declare the variables. $\Delta S_{i(prev)}$, ΔS_i and $Step_i$ is all set to 0 initially.

Then the program asks the users to input 8 numbers representing the length change for 8 cables. The 8 values will be assigned to variable $\Delta S_1 \dots \Delta S_8$ in order. During the input process, the user can input ‘clear’ instead of numbers to clear all the previous input and start over again from ΔS_1 .

Once the input process is complete, the program converts the ΔS_1 ΔS_8 to $Step_1$ $Step_8$ based on Equation 50, with the d set to 10mm, N set to 2048 steps/rev. The motors will start to step immediately after the conversion is finished.

During the stepping process, the motors controlling the first and second unit (the first 4 motors) are set to have a stepping speed of 300 steps/second, and the motors controlling the third and fourth unit (the last 4 motors) are set to have $\frac{300}{17.5} \cdot (17.5 + 15) \approx 557$ steps/second. This is because the moving speed for the last two units should be $\frac{17.5+15}{17.5}$ times faster than the first two units so that the 4 units can finish moving approximately at the same time.

The program shall wait until the motors finished stepping. Once the stepping process is finished, the program asks the user what to do next: reset the motors to initial position or move from the current position to the next position. If the user chooses to reset the motors, all of the motors rotate backwards for same steps. However, If the user chooses to let the motors move to a new position, the values for current $\Delta S_1 \Delta S_8$ will be stored to variable $\Delta S_{1(prev)} \Delta S_{8(prev)}$. The program then asks for a new set of $\Delta S_1 \Delta S_8$, then calculate the new $Step_1 Step_8$. Hence, the loop of the program is closed.

3.5.2 Parameter input by Serial Monitor

In the operating logic, the control parameters are inputted through the keyboard to control the motion of the multiple stepper motors. The basic function is to input the steps into the serial monitor through the keyboard of the host computer and the input string will be read by the microcontroller and converted into data, which are used as the number of steps of the stepper motor movement. For easy operation and display, the input parameters need to be displayed as text on the screen of the LCD IIC. LCD IIC will be in the form of a 2-line, 16-bit display to show more content.

Logic flowchart of the code to control the motor and LCD IIC by inputting parameters with the serial monitor of the host computer is shown in Figure 10a.

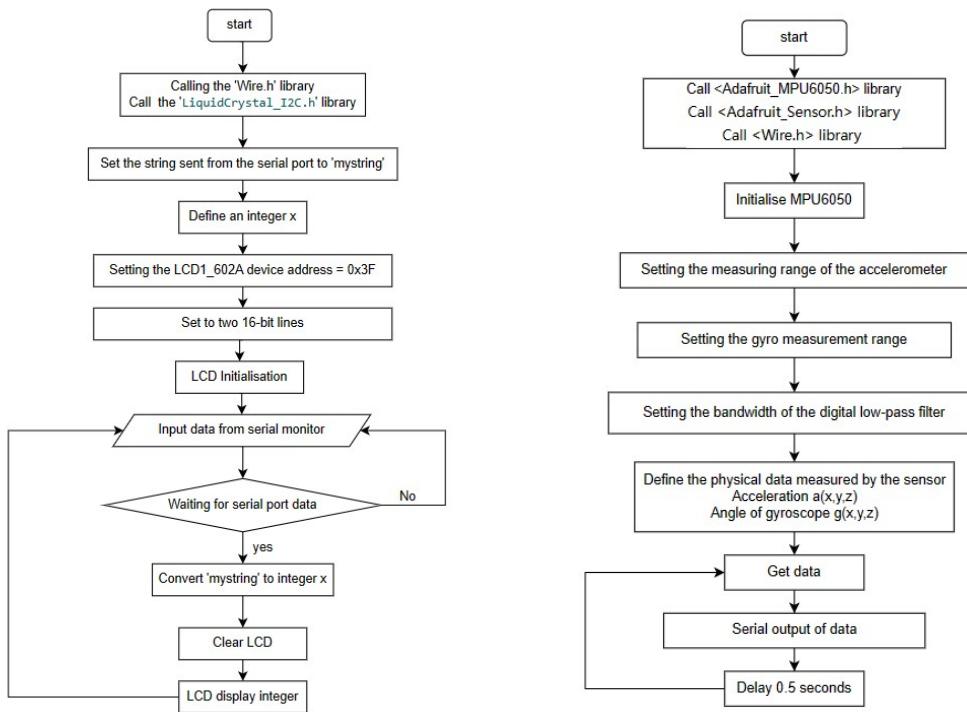
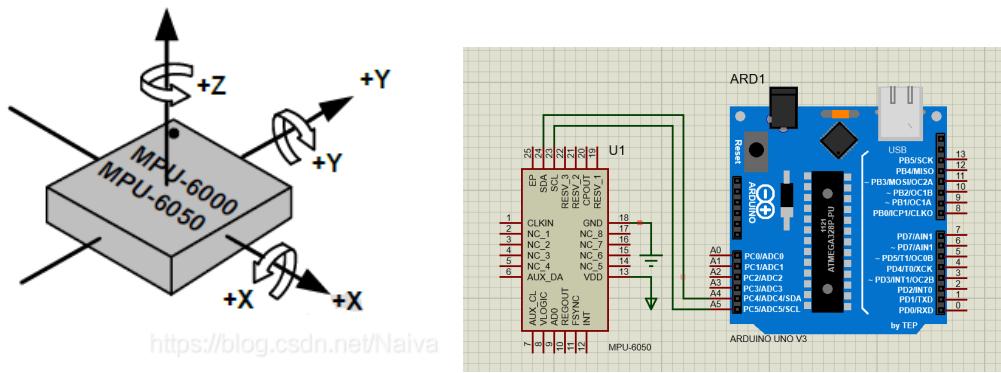


Figure 10: The code logic of input parameters and MPU6050.

3.5.3 Verification of MPU 6050

According to the project content, adding sensors to the end effector to measure its orientation angle and distance from the contact plane can assist the user to carry out more accurate operation. The function of detecting the orientation angle is realised by the MPU6050.

The Orientation angle (Euler angle) corresponding to the MPU6050 chip is shown in Figure 11a. Pitch is rotation around the x-axis. Yaw is rotation around the y-axis. Roll is rotation around the z-axis.



(a) The MPU6050 chip corresponding to (b) The wiring diagram of MPU6050 and the Euler angle microcontroller

Figure 11: The MPU6050 chip with ancillary Arduino circuit.

The MPU6050 requires an external 5V power supply and GND. SCL is the Clock Pin of the IIC when the MPU6050 is used as a slave, and SDA is the Data Pin of the IIC when the MPU6050 is used as a slave. The wiring diagram is shown in Figure 11b.

The logic flowchart of the code that controls the MPU 6050 to display acceleration and direction angle is shown in Figure 10b. In this project, the acceleration parameters will not be acquired and displayed.

Since Proteus does not support dynamic simulation of the MPU6050, the verification of the MPU6050 has been carried out by physical parts and Arduino Mega while the project is in progress. The validation results have an acceptable accuracy and are able to display the corresponding orientation angle on the serial port and OLED screen.

3.5.4 Using HC-SR04 to measure distance

The HC-SR04 Ultrasonic Sensor uses sonar to determine the distance to an object. It has an ultrasonic transmitter and a receiver module which provides excellent non-contact range detection, high accuracy and stable readings. The measurement range is from 2 - 400 cm. The resolution is 0.3 cm and the measurement angle is within 30°. It operates in a process unaffected by sunlight or ferrous materials.

The sensor is triggered by sending a HIGH pulse of 10 microseconds. Previously, the speaker gives a short low-level pulse to ensure that a clean HIGH pulse is obtained.

The basic principle of distance measurement is time of flight (Tof). The formula is as

follows.

$$distance = (duration/2) \times v_{sound} \quad (51)$$

$$v_{sound} = 343 \text{ m/s} = 1/29.1 \text{ cm/us}$$

The wiring diagram and simulation results are shown in Figure 12. An SSD 1306 OLED screen is used to display the data in the simulation. The logic flowchart of the code that controls the HC-SR04 to measure distance and display acceleration by OLED screen is shown in Figure 13a.

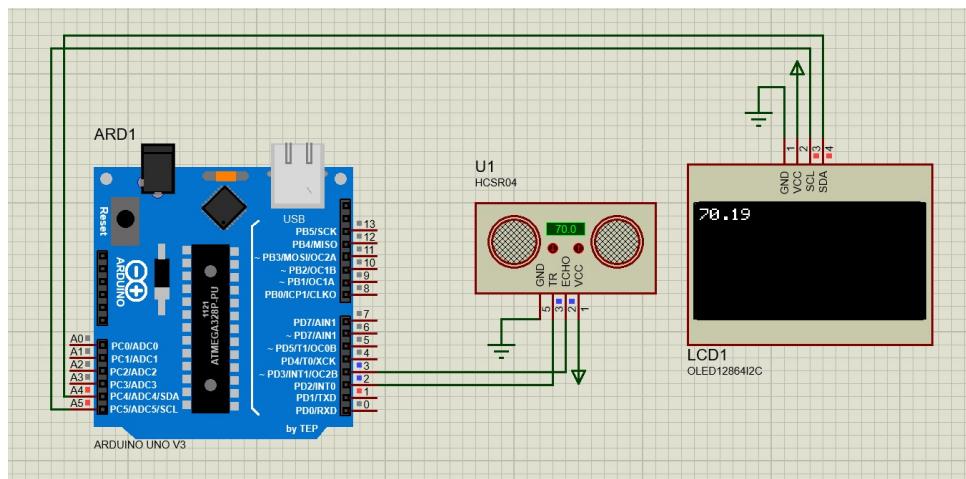
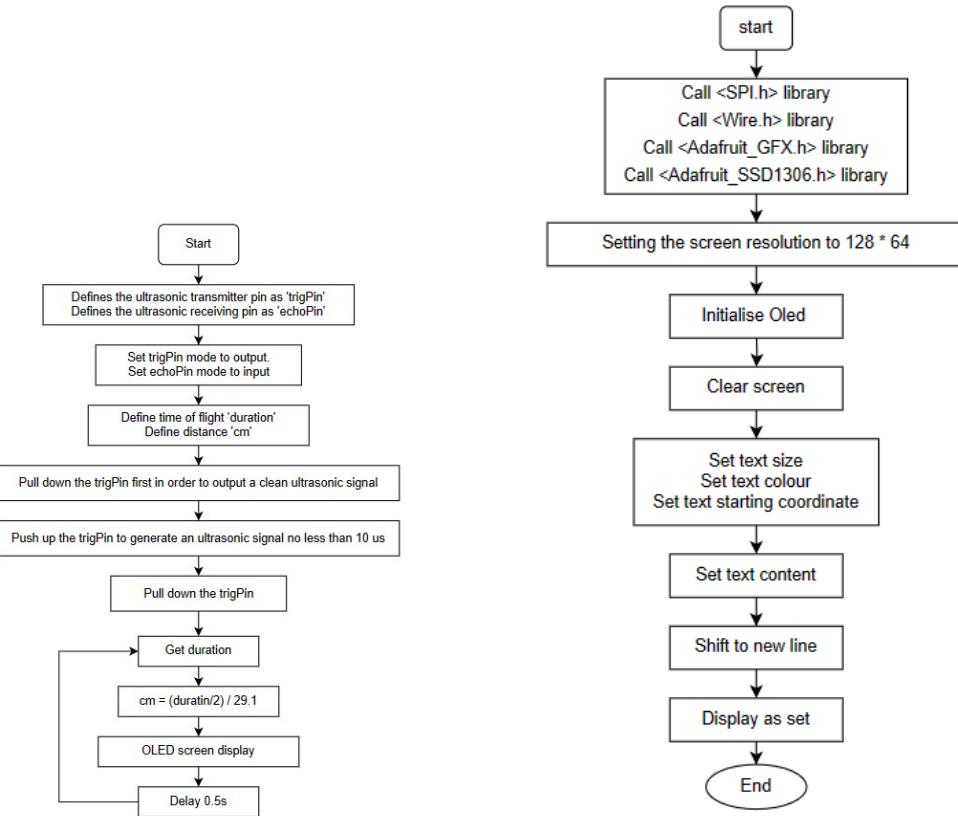


Figure 12: The wiring diagram and simulation of HC-SR04.



(a) The code logic of using HC-SR04 to measure distance (b) The code logic of using SSD 1306 OLED to display data

Figure 13: The code logic of HC-SR04 and SSD 1306 OLED.

3.5.5 SSD 1306 OLED

Using SSD 1306 OLED requires calling two libraries, `<Adafruit_SSD1306.h>` and `<Adafruit_GFX.h>`. `<Adafruit_SSD1306.h>` is a dedicated display library for SSD1306 OLED screens. `<Adafruit_GFX.h>` library is the common parent graphics library for LCD and OLED screens.

The resolution of SSD1306 is 128x64 pixel dot array, SCK is the clock pin, SDA is the data pin, and external power supply and GND are required. Figure 13b is the logic flowchart of the code to control the OLED to display the text content.

4 Result and Discussion

4.1 Error Analysis of Cross-Shaped Sheets

At the early stage of the project, given that the number of cross-shape sheets in each modules may have an influence on the maximum bending angle α_{max} , which is an essential parameter of those which affect the range of working space of the manipulator, hence the number of sheets should be determined by deriving the functions of both absolute error and relative error [3] against the number of sheets. The simulation results are shown in Figure 14.

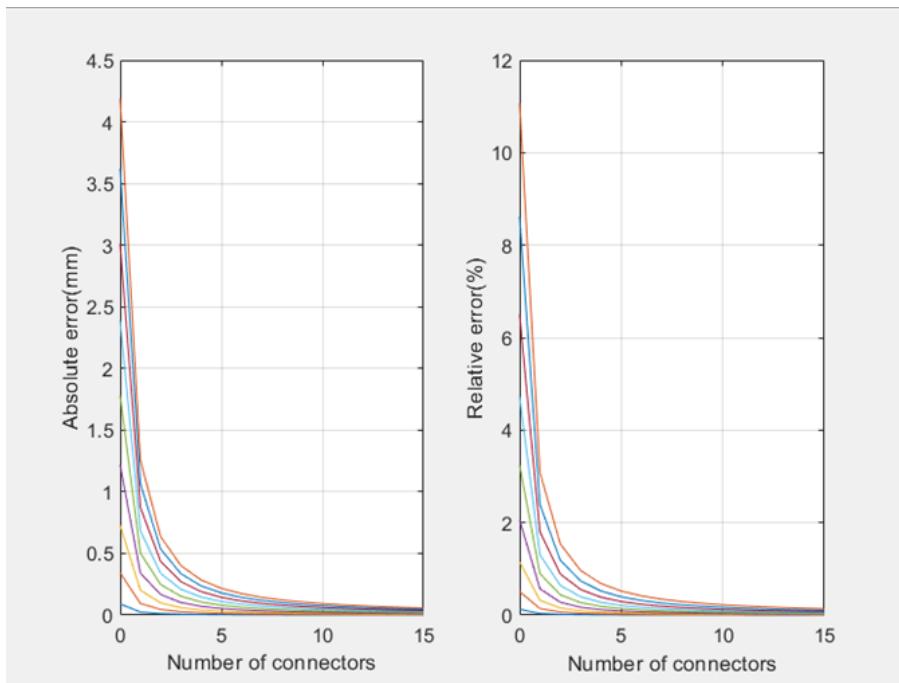


Figure 14: The simulation results of errors against the number of connecting sheets.

As illustrated in graphs, the changing trends of both absolute and relative errors are almost identical since the relative error is proportional to the absolute error, and the different lines with different colors represent the data with different bending angles whose range is from 10 degrees to 90 degrees in both directions (positive or negative). It is obvious that both errors decrease with the increase in the number of connecting sheets, hence theoretically, the number of sheets should be as large as possible to make both errors could be negligible. However, the number of sheets also has an influence on the size of simulation space, more the number of sheets less the size of simulation space

given that if the length of manipulator is much longer, the space near to the basement of manipulator could not be reached due to the characteristics of the material. Hence, the number of sheets could be assumed to be 10 rather than 15, even though both the errors are smaller when the number of sheets is 15.

4.2 Workspace Analysis of Manipulator

The simulation of workspace of the manipulator has been illustrated since it is important to determine the variables of joints based on the required workspace. The dimension of workspace is affected by several aspects, which include the simulation index and the parameter H representing the height of the basement of cubic in the original coordinate of the manipulator. The simulation results and the parameters are shown below.

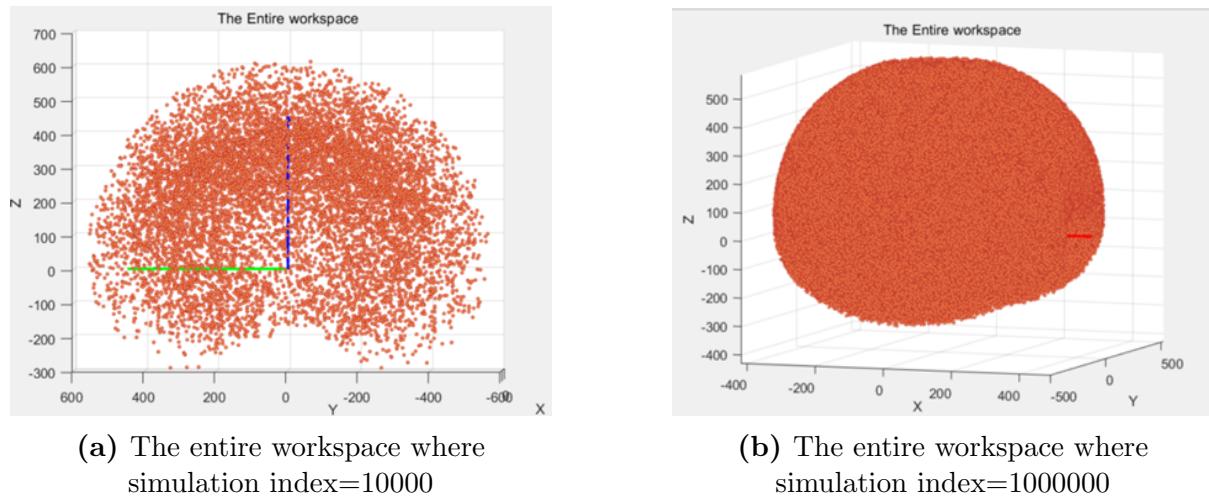
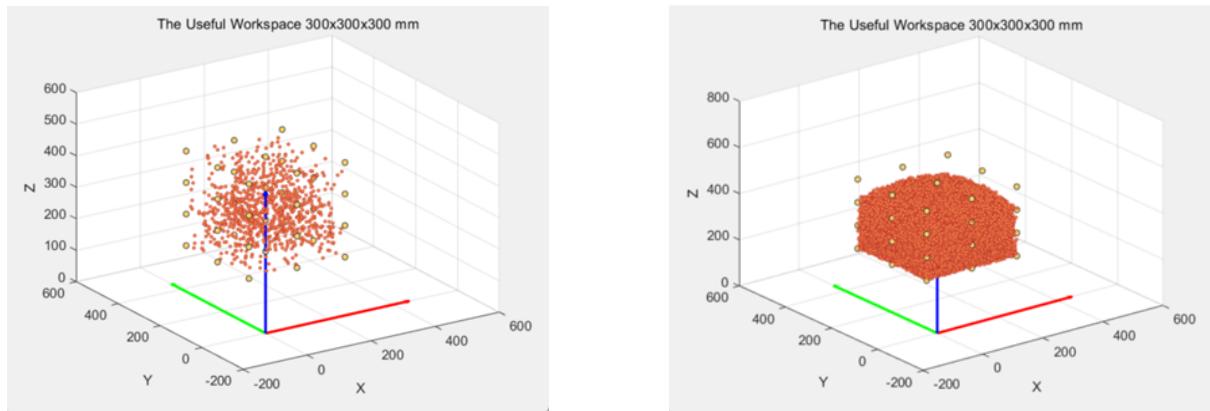


Figure 15: The entire workspace with different simulation indices.

As shown in Figure 15a, the working space of manipulator forms a spherical shell-shaped point cloud consisting of 10000 floating points since the simulation index is 10000, which is insufficient to fully occupied the theoretical workspace. To further generalize the theoretical workspace, it is necessary to enhance the simulation index. The appropriate workspace should be divided from the workspace.



(a) The cubic workspace of manipulator with index = 10000 H=260-560mm

(b) The cubic workspace of manipulator with index = 1000000 H=350-650mm

Figure 16: The cubic workspace with different simulation indices and different segmentation.

According to Figure 16, due to the limitation in the dimension of entire workspace, the upper part of the cubic space could not be fully occupied, in other words, is almost empty when $H=350\text{mm}$, which means that not all the points shown in yellow could be reached by the manipulator. Aiming to determine the ‘bond’ value of parameter H , the function of calculating the distance between the points in yellow in the same layer and the nearest floating points has been carried out and the simulation results with index=1000000 and $H=300\text{mm}$ are shown below.

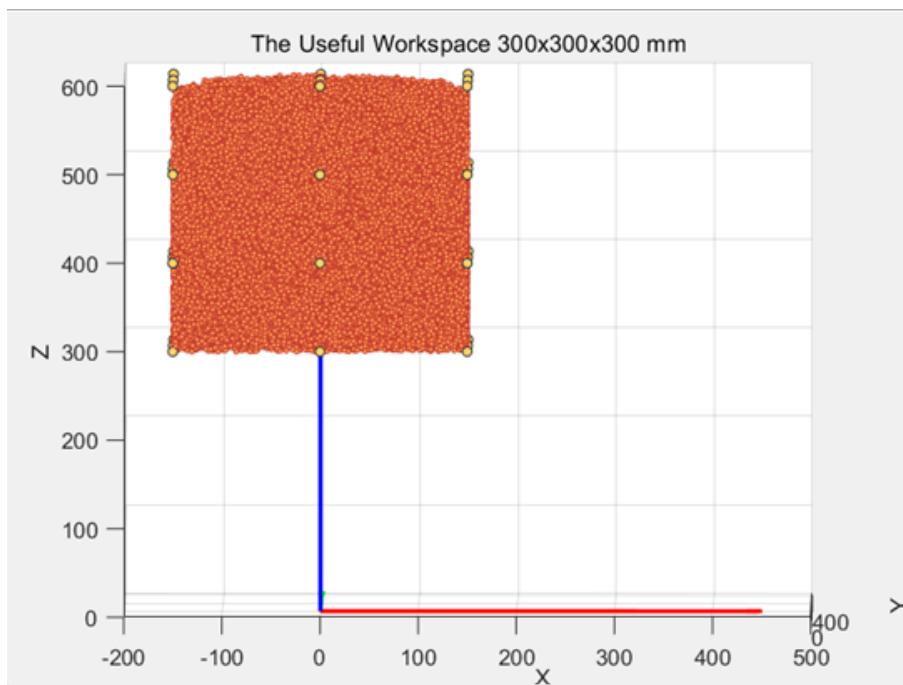


Figure 17: The cubic workspace in X-Z plane with H=300mm.

As shown in Figure 17, almost the entire cubic workspace has been occupied by the floating points which indicates that the value of H could be considered as 300mm, but it is not rigor to concluded that the ‘bond’ value is 300. The detailed data which shows the distance from the floating points to the nearest point which sits on the edge for each layer is shown below with the unit ‘mm’.

Height:300 mm									
7.0696	3.3457	5.8420	5.1691	6.9859	9.6064	12.0161	3.1330	8.0543	
Height:400 mm									
3.2065	5.6137	6.2471	1.9011	3.2564	6.4037	7.3970	5.9186	7.0688	
Height:500 mm									
10.3989	3.3835	9.2886	7.4610	3.5080	5.3177	9.7733	6.1874	1.8141	
Height:600 mm									
33.8095	13.3727	30.6118	6.0586	9.5076	8.8049	30.4824	16.8214	32.5093	

Figure 18: The distance between random points and the nearest points on the edge with H=300mm.

Before analyzing these data, the following top-down view of the cubic workspace will help to link each numerical value with the designated points.

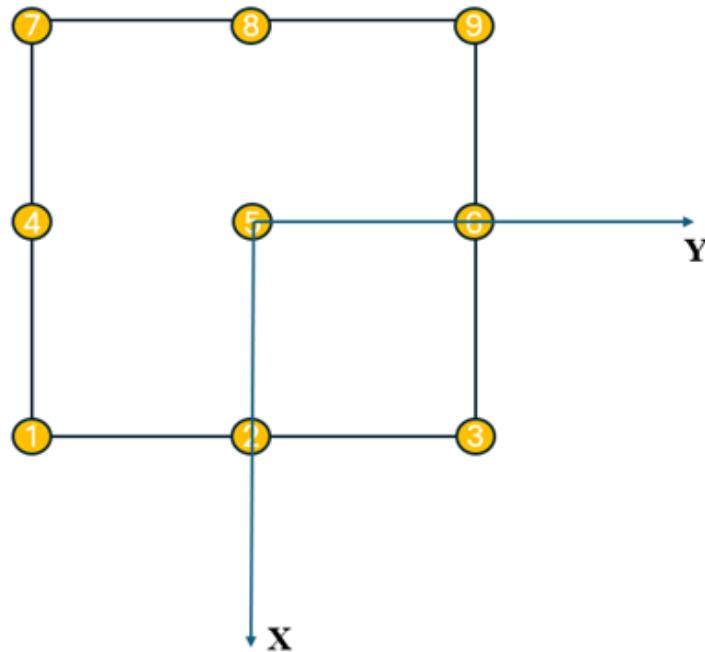


Figure 19: The top-down view of the cubic workspace.

The first layer is the one with height of 300mm and the fourth layer is the one with height of 600mm, and the first thing should be check is the average distance between the random points and the nearest ‘edge’ point of layer two and layer three given that if the

18 points of these two layers could not be fully surrounded by the entire workspace, then those laying on the layer one and four couldn't be reached by the end-effector as well. The average distance for layer 2 and 3 could be obtained by the following calculation.

$$d_{ave,300mm} = \frac{\sum_2^3 d}{18} = 5.7859mm \quad (52)$$

where 2 and 3 indicate the values of distance listed in the second and third row in Figure 18 and denominator means there are 18 points in two layers in total. Based on the value shown in row one and row four, the following table could be developed.

Table 3: The Parameters of Manipulators.

Number	1	2	3	4	5	6	7	8	9
Layer 1	7.07	3.35	5.84	5.17	6.99	9.61	12.02	3.13	8.05
Layer 4	33.81	13.37	30.61	6.06	9.51	8.81	30.48	16.82	32.51

Given that the average distance is 5.79mm, hence if the distance obtained in layer 1 and 4 is larger than the average one, it means that the end-effector could not reach that point. From the table and the calculated average distance in layer 2 and 3, the following points could be drawn.

- Except No.2, No.4, No.8, all other points in both layers could not be reached by the end-effector since the distance is larger than the average value.
- The value of parameter H for this case is not suitable since almost all the ‘edge’ points on layer 1 and layer 4 could not be reached by the end-effector.

Hence, other values of H should be taken and compare the results with those with H=300mm. In this case, the value of H is taken as 280mm, and the simulation results are shown below.

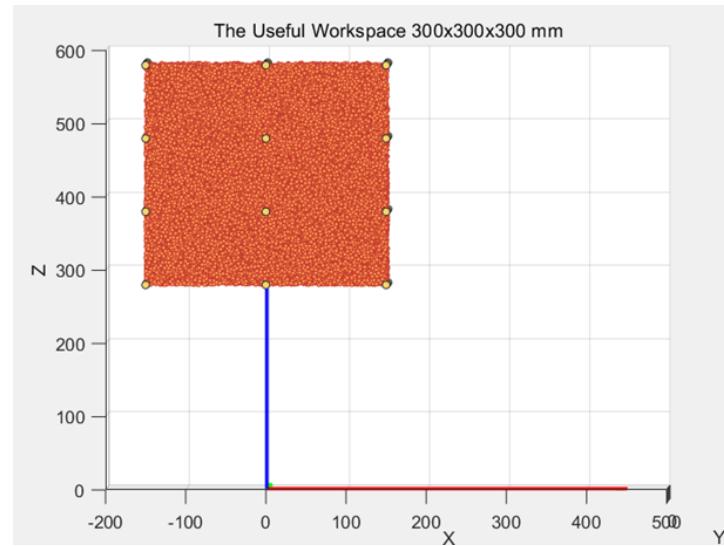


Figure 20: The cubic workspace with index=1000000, H=280mm.

Table 4: The Parameters of Manipulators.

H (mm)	240	260	280	300
Average distance (mm)	5.35	5.37	4.875	5.7859
Number of points could be reached in layer 1 and 4	1	4	5	3

Based on Table 4, the suitable value of parameter H, the ‘bond’ height of the cubic, could be set to 280mm since the number of points could be reached in layer 1 and 4 is 5 which is more ideal than other cases. However, there are still some limitations in deciding the value of H which are shown below.

- The value of the index is a little bit small, which has an influence on the distance between the floating points and the nearest ‘edge’ points. Hence the value of H could not be as precise as possible.
- There are too few trials have been carried out leading to the value of H is not convinced by others, and one possible way to deal with it is that develop a mathematic model i.e., Normal distribution, to predict the bond value of H which is more convincing.
- The data were obtained from few aspects which may affect the accuracy of the final value of H.

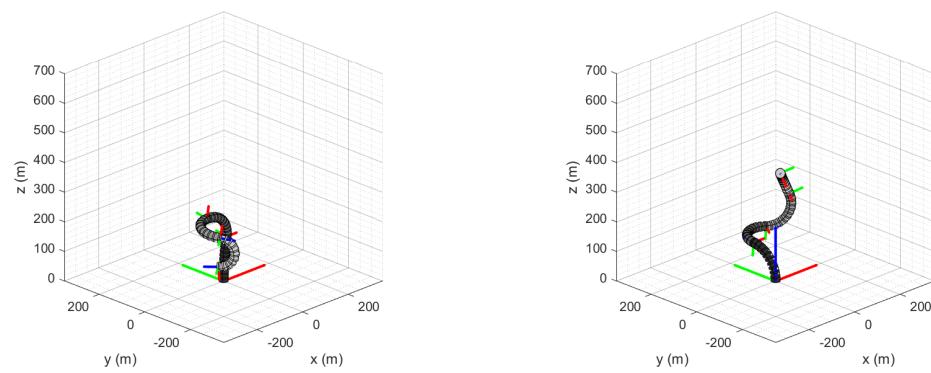
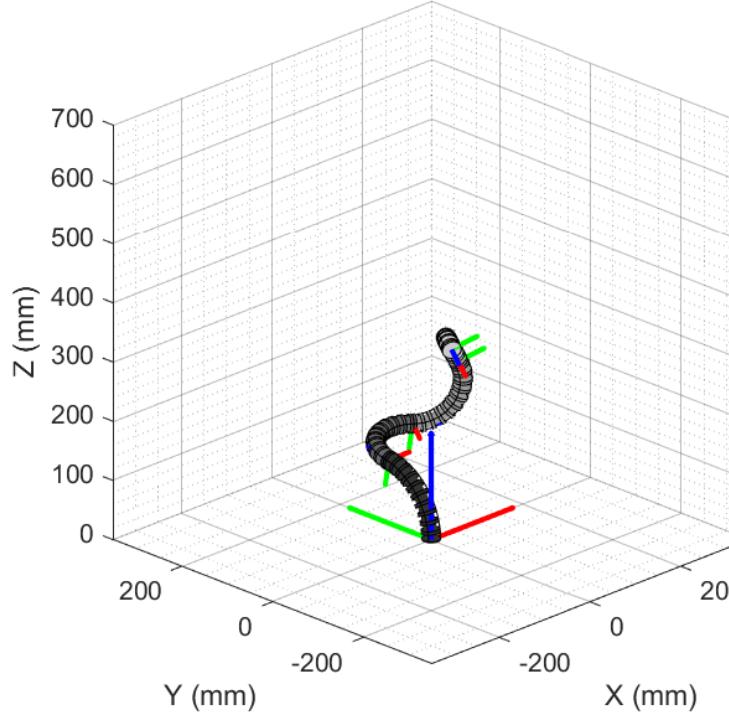
4.3 Inverse Kinematics

This section will demonstrate the singular posture solution and trajectory replication results of the inverse kinematics algorithm. A comprehensive analysis of its accuracy and computational speed will be conducted, along with a description of its advantages and limitations. To begin with, it is essential to note that using angles as input for the programme is feasible due to the relative complexity of specifying the \mathbf{O}_{target} and \mathbf{P}_{target} , which involves a total of 12 parameters. Moreover, the targets are generated based on angles by forward kinematics algorithm during the test. In essence, there is no substantial distinction between inputting angles and inputting coordinates.

4.3.1 Singular Posture Solution

Firstly, the bending of individual modules was tested as the target for the inverse kinematics solution. The inverse kinematics algorithm was tested for different angles of bending for Module 1, 2, 3, and 4, and the specific results are presented in Table A.2 in Appendix A. The solutions about the Modules 3 and 4 are highly satisfactory, with Module 3 even converging to error of 0.02mm in 6 epochs. However, the bending results for Module 1 and 2 are unsatisfied, which is caused by inappropriate initialization. According to the flowchart in Figure 7, the FABRIKc algorithm requires an initialization to start iterations. Taking the example of the target with $\alpha = [0, 90, 0, 0]^\circ$, after initializing with the initial posture in Figure 5 and applying the FABRIKc algorithm for iterations, the first iteration yields a posture with $\theta = [-0.0, 104.04, -0.0, -22.0]^\circ$. Moreover, since each iteration starts from Module 4, the error in Module 4 cannot be eliminated and persists. To mitigate this effect, selecting a more suitable posture for initialization is crucial, which can be solved trajectory replication. If the posture used for initialization with $\alpha = [0, 100, 0, 0]^\circ$, the solution of inverse kinematics algorithm would be $\theta = [-0.0, 81.77, -0.0, 9.3]^\circ$. Nevertheless, due to errors generated by the iteration sequence, complete elimination remains challenging, and efforts are focused on minimizing them as much as possible. Afterwards, the complex bending angles of modules was utilized to investigate the influences of initialization. The target with angles $\alpha = [80, 120, -120, 90]^\circ$ is selected because its solution is complex, and it lies within the workspace of manipulator. The

posture of the manipulator is shown in Figure 21a. This scenario is likely to occur in practical applications and necessitates resolution through relevant methods.



(b) initialization: $\alpha = [0, 0, 0, 0]^\circ$
 $\theta = [-8.88, -89.99, -126.81, -169.73]^\circ$

(c) initialization: $\alpha = [85, 125, -115, 55]^\circ$
 $\theta = [84.98, 122.93, -114.9, 63.01]^\circ$

Figure 21: The inverse kinematics algorithm with different initialization.

With the initialization using the initial posture, the solution is $\theta = [-8.88, -89.99, -126.81, -169.73]^\circ$, which is shown in Figure 21b. However, the solution of inverse kinematics algorithm would be $\theta = [84.98, 122.93, -114.9, 63.01]^\circ$, which is significantly close to the target while the angles for initialization are $[85, 125, -115, 55]^\circ$. The solution of the inverse kinematics with more appropriate

initialization is shown in Figure 21c. This signifies that if the manipulator is presently in a posture similar to the target configuration, it can accurately determine the corresponding bending angles θ through the inverse kinematics algorithm. This proves to be highly beneficial for trajectory replication.

Ultimately, The efficient computational capability of algorithm is one of its strengths. The algorithm only takes 1.905 seconds to complete 10,000 epochs updating. In practical applications, it requires only 200 epochs to determine convergence. In comparison to traditional methods like inverse Jacobian [62], which involve matrix transformations and derivatives, this approach can provide potential solutions in a short time, addressing the issue of singular points. However, this method has its limitations. In scenarios with multiple segments, the algorithm may introduce significant errors due to the calculation order, and these errors can be challenging to eliminate. The only viable solution is to enhance the algorithm's performance through careful initialization methods.

4.3.2 Trajectory Replication

The preceding discussion has highlighted the use of the current posture for initialization in trajectory replication. In this phase, two types of trajectories were replicated: arc segments and closed paths. This section will analyze these two trajectories, elucidating the strengths and weaknesses of inverse kinematics algorithm.

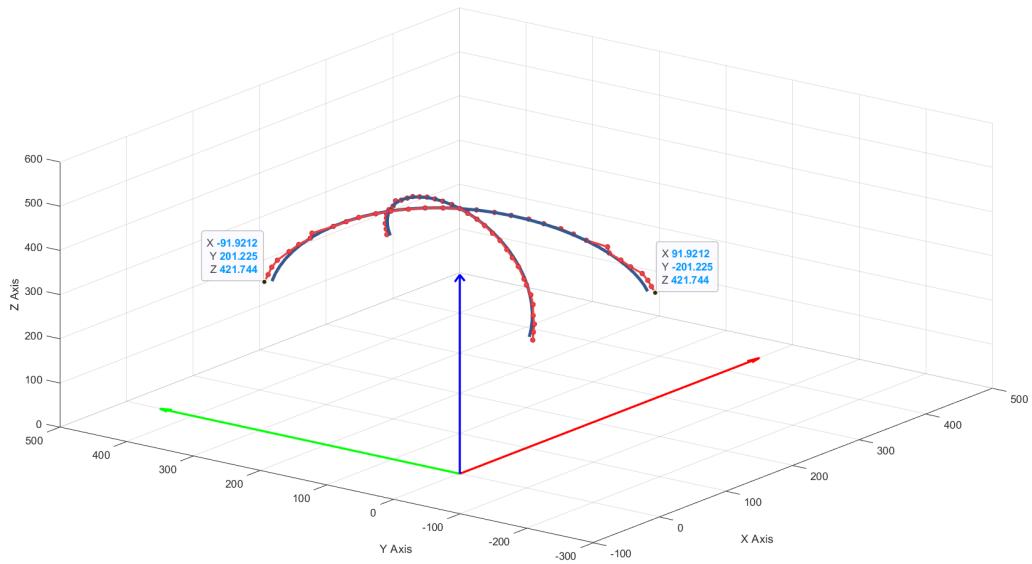


Figure 22: The cross-shaped trajectory and its replication by FABRIKc algorithm.

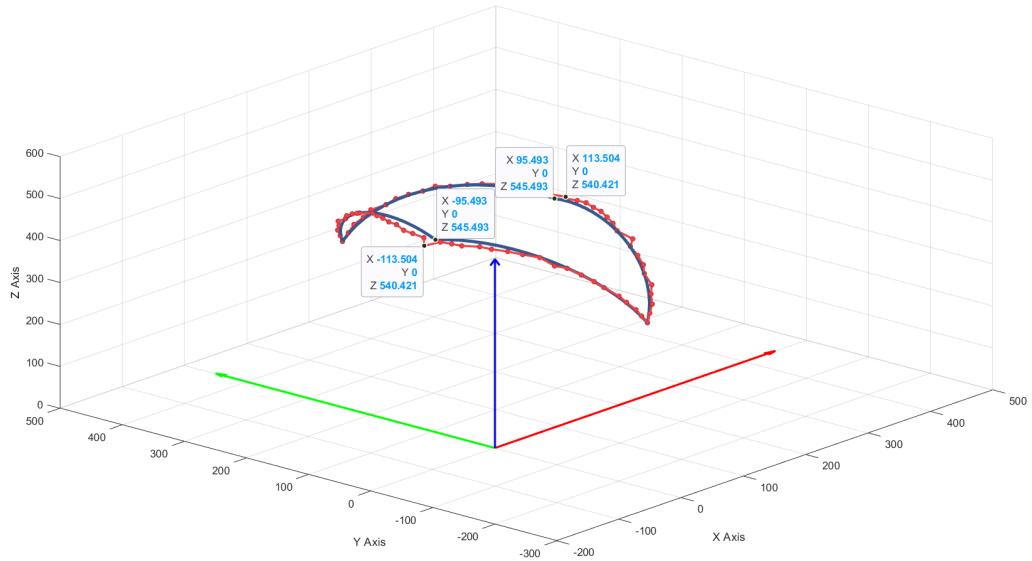


Figure 23: The closed trajectory and its replication by FABRIKc algorithm.

TODO: make analysis about this part, calculate the error

4.3.3 The improvement about Inverse kinematics algorithm

TODO:talking about improvement to do about inverse kinematics, which are more detailed control method

4.4 Electroinc Control

Initially, the testing of the results was supposed to be carried out through Proteus simulation. However, it was discovered that during the simulation process in Proteus, the motor components did not function normally, with frequent occurrences of stepping errors. At the same time, the simulation couldn't be completed in real-time due to the heavy load on the computer CPU. After consideration, the team members purchased a complete set of real components for testing. In the real circuit, the pin that each motor occupies are listed in the table below to reduce the complexity of wiring:

Connect the components as shown in the figure below, and run the program.

Once started, the first thing the user needs to do is to input the values of $\Delta S_1 \sim \Delta S_8$:

From the graph above we can see that $\Delta S_1 \sim \Delta S_8$ is:

0mm, 0mm, -8mm, 8mm, -15mm, 15mm, 65mm, -65mm.

The value of ΔS_i is always in pairs, e.g, two cables controlling the same unit should always have the opposite ΔS_i value, this is decided by the working principle of the manipulator.

After conversion, the corresponding $Step_1 \sim Step_8$ should be:

0, 0, -521, 521, -978, 978, 4237, -4237. (“-” sign represents counterclockwise steps).

Wait for the motors to finish stepping, once it's done, use function `Serial.print(motor.currentPosition())` to check if the motors stepped correctly:

As Figure above shows, all the motors are working correctly. In this program, all floating-point precision is taken to the last four decimal places (but the result printed on the monitor only shows two decimal places), so the accuracy of the result is beyond acceptable (from the above figure, maximum 0.01mm error).

Then, reset the motors, the motors went back to initial condition, which is $Step_1 \sim Step_8 = 0$. Repeat the motor stepping process, but this time, the motors are starting from a previous location from

0mm, 0mm, -8mm, 8mm, -15mm, 15mm, 65mm, -65mm to

33mm, -33mm, 16mm, -16mm, 0mm, 0mm, 10mm, -10mm.

The difference $\Delta S_i - DeltaS_i(prev)$ is:

33mm, -33mm, 24mm, -24mm, 15mm, -15mm, -55mm, 55mm.

The steps for motors are: 2151, -2151, 1565, -1565, 978, -978, -3585, 3585 steps.

Verify the result again by checking how many revolutions each motors rotated:

The results are correct. Hence, the actuation control part runs correctly. The program code is uploaded to Github:

<https://github.com/yezehao/Compact-Continuum-Manipulator-Platform/tree/main/Arduino-Simulation>

5 Conclusion

This is the Conclusion of the final report.

References

- [1] Sarthak Misra, K. T. Ramesh, and Allison M. Okamura. Modeling of tool-tissue interactions for computer-based surgical simulation: A literature review. *Presence*, 17(5):463–491, 2008.
- [2] Jessica Burgner-Kahrs, D. Caleb Rucker, and Howie Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.
- [3] Pan Zhou, Jiantao Yao, Shuai Zhang, Chunjie Wei, Hongyu Zhang, and Shupeng Qi. A bioinspired fishbone continuum robot with rigid-flexible-soft coupling structure. *Bioinspiration & Biomimetics*, 17(6):066012, 2022.
- [4] Jonathan M Sackier, Chuck Wooters, Lisa Jacobs, Amy Halverson, Darrin Uecker, and Yulun Wang. Voice activation of a surgical robotic assistant. *The American Journal of Surgery*, 174(4):406–409, 1997.
- [5] Satyam Kalan, Sanket Chauhan, Rafael F Coelho, Marcelo A Orvieto, Ignacio R Camacho, Kenneth J Palmer, and Vipul R Patel. History of robotic surgery. *Journal of Robotic Surgery*, 4:141–147, 2010.
- [6] TOMMASO FALCONE, JEFFREY GOLDBERG, Antonio Garcia-Ruiz, HAROUT MARGOSSIAN, and LAUREL STEVENS. Full robotic assistance for laparoscopic tubal anastomosis: a case report. *Journal of Laparoendoscopic & Advanced Surgical Techniques*, 9(1):107–113, 1999.
- [7] Hermann Reichenspurner, Ralph J Damiano, Michael Mack, Dieter H Boehm, Helmut Gulbins, Christian Detter, Bruno Meiser, Reinhard Ellgass, and Bruno Reichart. Use of the voice-controlled and computer-assisted surgical system zeus

- for endoscopic coronary artery bypass grafting. *The Journal of thoracic and cardiovascular surgery*, 118(1):11–16, 1999.
- [8] Bertrand Guillonneau, Olivier Cappèle, Juan Bosco Martinez, Stéphane Navarra, and Guy Vallancien. Robotic assisted, laparoscopic pelvic lymph node dissection in humans. *The journal of urology*, 165(4):1078–1081, 2001.
 - [9] F. Pugin, P. Bucher, and P. Morel. History of robotic surgery : From aesop® and zeus® to da vinci®. *Journal of Visceral Surgery*, 148(5, Supplement):e3–e8, 2011. Robotic surgery.
 - [10] Chih Yu An, Jia Hao Syu, Ching Shio Tseng, Chih-Ju Chang, et al. An ultrasound imaging-guided robotic hifu ablation experimental system and accuracy evaluations. *Applied Bionics and Biomechanics*, 2017, 2017.
 - [11] Sunita Chauhan, Hagey Amir, Guang Chen, Axel Hacker, Maurice Stephan Michel, and Kai Uwe Koehrmann. Intra-operative feedback and dynamic compensation for image-guided robotic focal ultrasound surgery. *Computer Aided Surgery*, 13(6):353–368, 2008.
 - [12] Sunita Chauhan. A mechatronic system for non invasive treatment of the breast cancers. In *Mechatronics and Machine Vision: John Billingsly (Eds)*. Research Studies Press Ltd, 2002.
 - [13] Sunita Chauhan, Ming Yeong Teo, and Wendy Teo. Robotic system for ablation of deep-seated skull base cancers—a feasibility study. In *Proceedings of the 34th International MATADOR Conference: Formerly The International Machine Tool Design and Conferences*, pages 21–27. Springer, 2004.
 - [14] Mohamed K Almekkaway, Islam A Shehata, and Emad S Ebbini. Anatomical-based model for simulation of hifu-induced lesions in atherosclerotic plaques. *International Journal of Hyperthermia*, 31(4):433–442, 2015.
 - [15] Laure-Anaïs Chanel, Florent Nageotte, Jonathan Vappou, Jianwen Luo, Loïc Cuville, and Michel de Mathelin. Robotized high intensity focused ultrasound (hifu) system for treatment of mobile organs using motion tracking by ultrasound imaging: An in vitro study. In *2015 37th Annual International Conference of*

the IEEE Engineering in Medicine and Biology Society (EMBC), pages 2571–2575. IEEE, 2015.

- [16] Andrea Cafarelli, Marco Mura, Alessandro Diodato, Andrea Schiappacasse, Matteo Santoro, Gastone Ciuti, and Arianna Menciassi. A computer-assisted robotic platform for focused ultrasound surgery: Assessment of high intensity focused ultrasound delivery. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1311–1314. IEEE, 2015.
- [17] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. John Wiley & Sons, 2020.
- [18] Michael E Moran. Evolution of robotic arms. *Journal of robotic surgery*, 1(2):103–111, 2007.
- [19] Mark E Rosheim. *Robot evolution: the development of anthrobotics*. John Wiley & Sons, 1994.
- [20] Bhaskar Dasgupta and TS1739334 Mruthyunjaya. The stewart platform manipulator: a review. *Mechanism and machine theory*, 35(1):15–40, 2000.
- [21] Ray L Page. Brief history of flight simulation. *SimTecT 2000 proceedings*, pages 11–17, 2000.
- [22] Roger Bostelman, James Albus, Nicholas Dagalakis, Adam Jacoff, John Gross, et al. Applications of the nist robocrane. In *Proceedings of the 5th International Symposium on Robotics and Manufacturing*, volume 5, page 1, 1994.
- [23] Anastasia Puzatova, Pshtiwan Shakor, Vittoria Laghi, and Maria Dmitrieva. Large-scale 3d printing for construction application by means of robotic arm and gantry 3d printer: A review. *Buildings*, 12(11):2023, 2022.
- [24] Byung-In Kim, Sunderesh S Heragu, Robert J Graves, and Art St Onge. Clustering-based order-picking sequence algorithm for an automated warehouse. *International Journal of Production Research*, 41(15):3445–3460, 2003.

- [25] Ishak WI Wan, WH Kit, and MA Awal. Design and development of eggplant harvester for gantry system. *Pertanika Journal of Science & Technology*, 18(2), 2010.
- [26] Eberhard Abele, Matthias Weigold, and Stefan Rothenbücher. Modeling and identification of an industrial robot for machining applications. *CIRP annals*, 56(1):387–390, 2007.
- [27] Shin-ichi Matsuoka, Kazunori Shimizu, Nobuyuki Yamazaki, and Yoshinari Oki. High-speed end milling of an articulated robot and its characteristics. *Journal of materials processing technology*, 95(1-3):83–89, 1999.
- [28] Hoai Nam Huynh, Hamed Assadi, Edouard Rivière-Lorphèvre, Olivier Verlinden, and Keivan Ahmadi. Modelling the dynamics of industrial robots for milling operations. *Robotics and Computer-Integrated Manufacturing*, 61:101852, 2020.
- [29] See Han Tay, Wai Heng Choong, and Hou Pin Yoong. A review of scara robot control system. In *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, pages 1–6, 2022.
- [30] Rishabh Chaturvedi, Anas Islam, and Kamal Sharma. Anticipated investigation of a cylindrical robot arm by means of compound materials. *Eur J Mol Clin Med*, 7(4):736–745, 2020.
- [31] Foster Collins and Mark Yim. Design of a spherical robot arm with the spiral zipper prismatic joint. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 2137–2143. IEEE, 2016.
- [32] Varnita Verma, A Gupta, MK Gupta, and P Chauhan. Performance estimation of computed torque control for surgical robot application. *Journal of Mechanical Engineering and Sciences*, 14(3):7017–7028, 2020.
- [33] Erhan Akdoğan and Mehmet Arif Adli. The design and control of a therapeutic exercise robot for lower limb rehabilitation: Physiotherabot. *Mechatronics*, 21(3):509–522, 2011.

- [34] Lu Zongxing, Li Wanxin, and Zhang Liping. Research development of soft manipulator: A review. *Advances in Mechanical Engineering*, 12(8):1687814020950094, 2020.
- [35] Deepak Trivedi, Christopher D Rahn, William M Kier, and Ian D Walker. Soft robotics: Biological inspiration, state of the art, and future research. *Applied bionics and biomechanics*, 5(3):99–117, 2008.
- [36] Cecilia Laschi, Barbara Mazzolai, Virgilio Mattoli, Matteo Cianchetti, and Paolo Dario. Design of a biomimetic robotic octopus arm. *Bioinspiration & biomimetics*, 4(1):015006, 2009.
- [37] Michael W Hannan and Ian D Walker. Kinematics and the implementation of an elephant’s trunk manipulator and other continuum style robots. *Journal of robotic systems*, 20(2):45–63, 2003.
- [38] Cecilia Laschi, Matteo Cianchetti, Barbara Mazzolai, Laura Margheri, Maurizio Follador, and Paolo Dario. Soft robot arm inspired by the octopus. *Advanced robotics*, 26(7):709–727, 2012.
- [39] Fionnuala Connolly, Conor J Walsh, and Katia Bertoldi. Automatic design of fiber-reinforced soft actuators for trajectory matching. *Proceedings of the National Academy of Sciences*, 114(1):51–56, 2017.
- [40] Shujun Wei, Tianyu Wang, and Guowei Gu. Design of a soft pneumatic robotic gripper based on fiber-reinforced actuator. *Journal of Mechanical Engineering*, 53(13):29–38, 2017.
- [41] Bobak Mosadegh, Panagiotis Polygerinos, Christoph Keplinger, Sophia Wennstedt, Robert F Shepherd, Unmukt Gupta, Jongmin Shim, Katia Bertoldi, Conor J Walsh, and George M Whitesides. Pneumatic networks for soft robotics that actuate rapidly. *Advanced functional materials*, 24(15):2163–2170, 2014.
- [42] Ramses V Martinez, Jamie L Branch, Carina R Fish, Lihua Jin, Robert F Shepherd, Rui Nunes, Zhigang Suo, and George McClelland Whitesides. Robotic tentacles with three-dimensional mobility based on flexible elastomers. *Advanced materials*, 2013.

- [43] Run Wang, Nan Jiang, Jian Su, Qu Yin, Yue Zhang, Zhongsheng Liu, Haibao Lin, Francisco A Moura, Ningyi Yuan, Siegmar Roth, et al. A bi-sheath fiber sensor for giant tensile and torsional displacements. *Advanced Functional Materials*, 27(35):1702134, 2017.
- [44] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.
- [45] Caleb Christianson, Nathaniel N Goldberg, Dimitri D Deheyn, Shengqiang Cai, and Michael T Tolley. Translucent soft robots driven by frameless fluid electrode dielectric elastomer actuators. *Science Robotics*, 3(17):eaat1893, 2018.
- [46] QM Zhang, Hengfeng Li, Martin Poh, Feng Xia, Z-Y Cheng, Haisheng Xu, and Cheng Huang. An all-organic composite actuator material with a high dielectric constant. *Nature*, 419(6904):284–287, 2002.
- [47] Bertrand Tondu, Serge Ippolito, Jérémie Guiochet, and Alain Daidie. A seven-degrees-of-freedom robot-arm driven by pneumatic artificial muscles for humanoid robots. *The International Journal of Robotics Research*, 24(4):257–274, 2005.
- [48] David Jakes, Zongyuan Ge, and Liao Wu. Model-less active compliance for continuum robots using recurrent neural networks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2167–2173. IEEE, 2019.
- [49] Ernar Amanov, Thien-Dang Nguyen, and Jessica Burgner-Kahrs. Tendon-driven continuum robots with extensible sections—a model-based evaluation of path-following motions. *The International Journal of Robotics Research*, 40(1):7–23, 2021.
- [50] Ammar Amouri, Abdelhakim Cherfia, Ayman Belkhiri, and Halim Merabti. Bio-inspired a novel dual-cross-module sections cable-driven continuum robot: design, kinematics modeling and workspace analysis. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 45(5):265, 2023.
- [51] Tomer Anor, Joseph R. Madsen, and Pierre Dupont. Algorithms for design of continuum robots using the concentric tubes approach: A neurosurgical example.

In *2011 IEEE International Conference on Robotics and Automation*, pages 667–673, 2011.

- [52] Jorge G Cham, Sean A Bailey, Jonathan E Clark, Robert J Full, and Mark R Cutkosky. Fast and robust: Hexapedal robots via shape deposition manufacturing. *The International Journal of Robotics Research*, 21(10-11):869–882, 2002.
- [53] Andrew D Marchese, Robert K Katzschmann, and Daniela Rus. A recipe for soft fluidic elastomer robots. *Soft robotics*, 2(1):7–25, 2015.
- [54] Haili Li, Jiantao Yao, Pan Zhou, Xinbo Chen, Yundou Xu, and Yongsheng Zhao. High-force soft pneumatic actuators based on novel casting method for robotic applications. *Sensors and Actuators A: Physical*, 306:111957, 2020.
- [55] Ali Zolfagharian, Abbas Z Kouzani, Sui Yang Khoo, Amir Ali Amiri Moghadam, Ian Gibson, and Akif Kaynak. Evolution of 3d printed soft actuators. *Sensors and Actuators A: Physical*, 250:258–272, 2016.
- [56] Cosima Du Pasquier, Tian Chen, Skylar Tibbits, and Kristina Shea. Design and computational modeling of a 3d printed pneumatic toolkit for soft robotics. *Soft robotics*, 6(5):657–663, 2019.
- [57] Ammar Amouri, Abdelhakim Cherfia, Ayman Belkhiri, and Halim Merabti. Bio-inspired a novel dual-cross-module sections cable-driven continuum robot: design, kinematics modeling and workspace analysis. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 45(5):265, 2023.
- [58] Sébastien Briot, Wisama Khalil, Sébastien Briot, and Wisama Khalil. Homogeneous transformation matrix. *Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements*, pages 19–32, 2015.
- [59] Andreas Aristidou and Joan Lasenby. Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260, 2011.
- [60] Weihao Zhang, Zhixiong Yang, Tianlai Dong, and Kai Xu. Fabrikc: an efficient iterative inverse kinematics solver for continuum robots. In *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 346–352, 2018.

- [61] Arduino comparison guide - sparkfun learn. (Accessed on 03 March 2024).
- [62] Isuru S Godage, Emanuele Guglielmino, David T Branson, Gustavo A Medrano-Cerda, and Darwin G Caldwell. Novel modal approach for kinematics of multisection continuum arms. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1093–1098. IEEE, 2011.

Appendix A

The Appendix of Tables

Table A.1: The results of FABRIKc algorithm with target angle $\alpha = [0 \ 0 \ 90 \ 0]$.

Epoch	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	Error (mm)
1	6.71	-0.0	83.29	-0.0	33.77506
2	6.77	-0.0	83.23	-0.0	34.07841
3	6.50	-0.0	83.50	-0.0	32.69662
4	6.37	-0.0	83.63	-0.0	32.07427
5	6.32	-0.0	83.68	-0.0	31.79337
6	6.29	-0.0	83.71	-0.0	31.66917
7	5.94	-0.0	84.06	-0.0	29.88369
8	5.74	-0.0	84.26	-0.0	28.91057
9	5.66	-0.0	84.34	-0.0	28.4723
10	5.62	-0.0	84.38	-0.0	28.2791
11	5.6	-0.0	84.4	-0.0	28.19218
12	5.25	-0.0	84.75	-0.0	26.43303
13	5.1	-0.0	84.9	-0.0	25.64898
14	5.03	-0.0	84.97	-0.0	25.29275
15	5.0	-0.0	85.0	-0.0	25.13415
16	4.98	-0.0	85.02	-0.0	25.06414
17	4.63	-0.0	85.37	-0.0	23.31238
18	4.48	-0.0	85.52	-0.0	22.52919
19	4.41	-0.0	85.59	-0.0	22.17736

Continued on next page

Table A.1 – continued from previous page

Epoch	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	Error (mm)
20	4.38	-0.0	85.62	-0.0	22.01754
21	4.36	-0.0	85.64	-0.0	21.94852
22	4.01	-0.0	85.99	-0.0	20.19744
23	3.84	-0.0	86.16	-0.0	19.29379
24	3.76	-0.0	86.24	-0.0	18.8911
25	3.72	-0.0	86.28	-0.0	18.71167
26	3.7	-0.0	86.3	-0.0	18.63148
27	3.7	-0.0	86.3	-0.0	18.59747
28	3.35	-0.0	86.65	-0.0	16.86972
29	3.2	-0.0	86.8	-0.0	16.09738
30	3.13	-0.0	86.87	-0.0	15.7551
31	3.1	-0.0	86.9	-0.0	15.59885
32	3.09	-0.0	86.91	-0.0	15.53487
33	3.08	-0.0	86.92	-0.0	15.50114
34	3.08	-0.0	86.92	-0.0	15.48938
35	2.74	-0.0	87.26	-0.0	13.77029
36	2.59	-0.0	87.41	-0.0	13.00123
37	2.52	-0.0	87.48	-0.0	12.65954
38	2.47	-0.0	87.53	-0.0	12.43367
39	2.45	-0.0	87.55	-0.0	12.33092
40	2.44	-0.0	87.56	-0.0	12.28876
41	2.44	-0.0	87.56	-0.0	12.26754
42	2.44	-0.0	87.56	-0.0	12.25842
43	2.1	-0.0	87.9	-0.0	10.55236
44	1.95	-0.0	88.05	-0.0	9.79285
45	1.88	-0.0	88.12	-0.0	9.45857
46	1.85	-0.0	88.15	-0.0	9.3098
47	1.84	-0.0	88.16	-0.0	9.24442
48	1.83	-0.0	88.17	-0.0	9.21438
49	1.83	-0.0	88.17	-0.0	9.19955
50	1.83	-0.0	88.17	-0.0	9.19355

Continued on next page

Table A.1 – continued from previous page

Epoch	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	Error (mm)
51	1.49	-0.0	88.51	-0.0	7.48619
52	1.34	-0.0	88.66	-0.0	6.73267
53	1.27	-0.0	88.73	-0.0	6.39321
54	1.24	-0.0	88.76	-0.0	6.24293
55	1.23	-0.0	88.77	-0.0	6.17847
56	1.22	-0.0	88.78	-0.0	6.14882
57	1.22	-0.0	88.78	-0.0	6.13682
58	1.22	-0.0	88.78	-0.0	6.1309
59	1.22	-0.0	88.78	-0.0	6.12797
60	1.22	-0.0	88.78	-0.0	6.12521
61	0.88	-0.0	89.12	-0.0	4.42272
62	0.73	-0.0	89.27	-0.0	3.66632
63	0.66	-0.0	89.34	-0.0	3.31015
64	0.63	-0.0	89.37	-0.0	3.1559
65	0.61	-0.0	89.39	-0.0	3.08583
66	0.61	-0.0	89.39	-0.0	3.05386
67	0.61	-0.0	89.39	-0.0	3.03937
68	0.6	-0.0	89.4	-0.0	3.03352
69	0.6	-0.0	89.4	-0.0	3.03063
70	0.6	-0.0	89.4	-0.0	3.0305
71	0.6	-0.0	89.4	-0.0	3.03044
72	0.6	-0.0	89.4	-0.0	3.03041
73	0.6	-0.0	89.4	-0.0	3.0304
74	0.27	-0.0	89.73	-0.0	1.33635
75	0.12	-0.0	89.88	-0.0	0.59092
76	0.05	-0.0	89.95	-0.0	0.25938
77	0.02	-0.0	89.98	-0.0	0.11515
78	0.01	-0.0	89.99	-0.0	0.05169
79	0.0	-0.0	90.0	-0.0	0.02296

Table A.2: The Singular Posture Solution by FABRIKc start with initial posture.

α_1	α_2	α_3	α_4	Converge Epoch	θ_1	θ_2	θ_3	θ_4	Error
0	0	0	90	None	-0.0	3.34	-0.0	89.09	18.6954
0	0	0	60	None	-0.0	1.61	-0.0	59.58	9.5398
0	0	0	30	None	-0.0	0.49	-0.0	29.67	2.77844
0	90	0	0	79	0.0	-0.0	90.0	-0.0	0.02296
0	60	0	0	36	0.01	-0.0	59.99	-0.0	0.03408
0	30	0	0	6	0.01	-0.0	29.99	-0.0	0.02928
0	0	90	0	None	-0.0	89.84	-0.0	0.28	0.66891
0	0	60	0	None	-0.0	76.97	-0.0	-22.73	79.01594
0	0	30	0	None	-0.0	52.28	-0.0	-22.87	114.34898
90	0	0	0	None	106.26	-0.0	-16.26	-0.0	80.98985
60	0	0	0	None	82.04	-0.0	-22.04	-0.0	111.99102
30	0	0	0	None	52.72	-0.0	-22.72	-0.0	117.37369

Appendix B

Test

This is the Appendix 2.