BUILDING A ROBOTIC ARM TO AUTOMATE THE PROCESS OF PATIENT CARE "MEASURING VITAL SIGNS" DURING CORONAVIRUS PANDEMIC

Submitted by:

Adel Mostafa
Mahmoud Abdelrhman
Mohamed Mohsen
Mohamed Yasser
Omar Ibrahim

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
BACHELOR OF SCIENCE
in
Systems and Biomedical Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2021

BUILDING A ROBOTIC ARM TO AUTOMATE THE PROCESS OF PATIENT CARE "MEASURING VITAL SIGNS" DURING CORONAVIRUS PANDEMIC

Submitted by:

Adel Mostafa
Mahmoud Abdelrhman
Mohamed Mohsen
Mohamed Yasser
Omar Ibrahim

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
BACHELOR OF SCIENCE
in
Systems and Biomedical Engineering

Under the supervision of:
Prof. Dr. Muhammad M. Islam
Professor of Electronics
Systems and Biomedical Engineering Department
Faculty of Engineering, Cairo University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2021

Acknowledgments

For us to actually be able to produce this project, we had to ask and learn from a lot of people and this a simple show of gratitude and gratefulness towards their time and help.

First, we want to give our best of gratitude to our supervisor and mentor Dr. Muhammad Islam for supporting us all the way until the end of the project. He was here for us every step of the way until producing the project stated in the following pages to come. To Dr. Muhammad Islam, we say thank you for the long journey.

Second, we would like to thank Information Technology Industry Development Agency (ITIDA) for helping us in the fabrication process and giving us the opportunity to work in their workspaces.

Finally, we would like to thank our college and ITIDA for accepting our proposal and funding our project.

To all these people and the various others who helped us, we say Thank you.

Dedication

Every challenging work needs self-efforts as well as guidance of elders especially those who were very close to our heart.

My humble effort I dedicate to my sweet and loving father and mother. Their affection, love, encouragement, and prays of day and night make us able to get such success and honor.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

WHO: World Health Organization

DOF: Degrees of Freedom

PWM: Pulse Width Modulation

HCW: Healthcare Workers

SARS-COV: Severe Acute Respiratory Syndrome Coronavirus

DH: Denavit Hartenberg

Abstract

Statistics showed that the most endangered people to catch coronavirus were the medical staff. We found that (10-15) % of the Covid-19 total infected persons in all the world were among the medical staff according to World Health Organization (WHO) [1].

The purpose of this project was to build a fully controlled robotic arm that could reduce the risk of infectious diseases transmission especially coronavirus to frontline medical staff by making them able to perform their tasks such as measuring vital signs without being in a close distance with the patients.

A humanoid right robotic arm from plywood sheets consisted of four links of different lengths. The healthcare robotic arm had four Degrees of Freedom (DOF). The total arm length was 60 cm with four continuous servomotors. We simulated the arm by AUTOCAD at first to test the mechanism capabilities.

We inserted XYZ point of the position we wanted the arm to move. Using inverse kinematics in MATLAB simulation, we got some angles. We converted the resultant angles into Pulse Width Modulation (PWM) by using excel sheet to map the angles into PWM. The motors stopped at the wanted position using bang bang control.

The result was the arm could reach any inserted point within the range of the DOF.

In summary, we built a fully controlled prototype for a humanoid robotic arm to assist the medical staff that will help healthcare workers in the fighting against the outbreak of Covid-19 crisis by reducing the chance of dealing with Covid-19 patients.

Chapter 1: Introduction

1.1 Problem Definition

The coronavirus disease has no mercy in threatening people's lives all over the world. Nationally, statistics from WHO had shown that 282,582 confirmed cases with 16,332 deaths from 3 January 2020 until the end of June 2021 [1]. There are many Healthcare workers "HCW" in any healthcare facility who struggle against this pandemic as they are in a close contact with the infected individuals while performing their work. Covid-19 catches several HCWs and causes severe acute respiratory syndrome coronavirus "SARS-COV" and the result is death. Globally, over 10,000 HCWs in Africa caught coronavirus according to the WHO reports until September 2020. This number is still in an increase and reaches 570,000 cases in 37 nation -including Egypt- with 2500 dead HCWs. In Egypt, there are 750 infected HCWs with 56 dead ones [2].

1.2 Problem Solution

Putting masks, keeping physical distance and other extra precautions of personal hygiene to cope with the changes in our daily life because of the outbreak of covid-19 are not enough as HCWs have to contact with patients directly for example when taking a blood sample from the patient. Robots are a magical solution to reduce the risk of spreading coronavirus and to save many HCWs from catching the virus as we can depend on them as shown in Fig. 1 [3], [4].

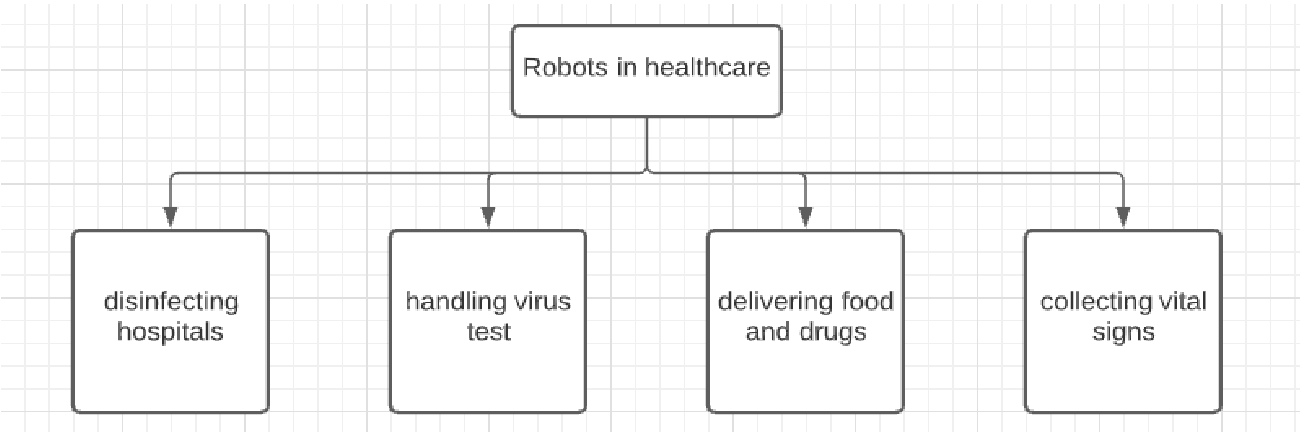

Figure 1: Robotic tasks in healthcare

1.3 Objective

We aim to build a robotic arm that related to what we face during the period of Covid-19 to assist the medical staff in doing some of their tasks in a fast and safe manner, thus not only protecting them from the virus but also reducing the burden on them. A healthcare robotic arm can significantly reduce the risk of infectious diseases transmission to frontline HCWs by making them able to perform their tasks without being in a close distance with the patients. Building a healthcare robotic arm can automate the process of patient care at any time especially during coronavirus pandemic directly and safely. We are mainly focused in the control of the arm which will make a lot of functions like any nurse such as measuring the patient's vital signs like patient's temperature and heart rate.

1.4 Project Overview

The block diagram describes our work as in Fig. 2. In this project, we build a right robotic arm with four motors that attached to a fixed body. First, we insert the XYZ point of the position I want move the arm to. Then we pass the point to inverse kinematics using MATLAB that produces angles as each motor has its angle. The motors moves with PWM, so we convert the angles into PWM. Bang-bang control is suitable for our motors to stop it at its certain position depending on the feedback of its magnetic encoder. After the previous steps, we move the motors to its destination and stop at it.

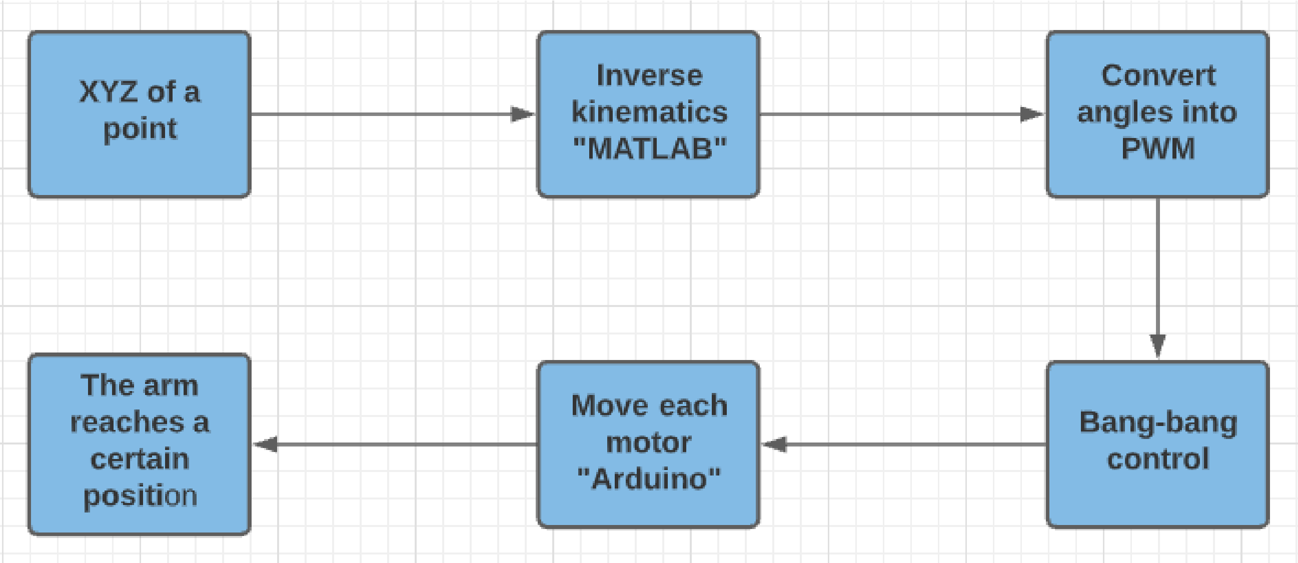

Figure 2: Block diagram represents the project

1.5 Gantt Chart to Implement Objectives

Fig. 3 shows a project management tool to illustrate our project plan.

Task	START DATE	END DATE	DURATION
Design of Arm and detect components	01/24	02/05	13
Fabrication and circuits connection	03/07	03/20	14
Start in software	03/21	04/17	28
Countinue in software and modify some parts in hardware	04/24	05/11	17
Finishing software, testing control of arm and starting in connection protocol	05/16	05/26	11
Writing the document	05/27	06/05	10
Finish the project and document	07/05	07/23	19

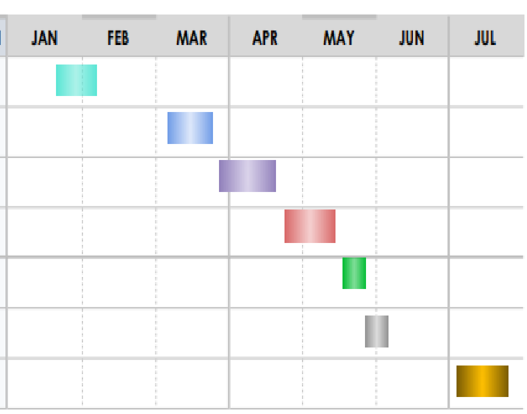

Figure 3: Gantt chart to implement objectives

Chapter 2: Market Research

- ***Target customers:*** Class-A hospitals as they seek for new technologies because they have the best medical staff.
- ***Competitors:*** Global healthcare robotics companies outside Egypt which we can export robotics from them.
- ***SWOT analysis:***

  Strengths:
  - The robotic arm can measure vital signs such as patient's temperature from a far distance.
  - It is easy to use.
  - It is a cheap product.
  - It decreases the risk of infectious diseases transmission to the medical staff

  Weaknesses:
  - The process of measuring may take time.
  - The robotic arm needs to patient interaction.
  - Few mentalities to trust robotics.
  - It will be a new product and needs for marketing.

  Opportunities:
  - The period of infectious diseases such as COVID-19.
  - The need to reduce the burden on the medical staff to accomplish much important tasks.
  - Our country seeks to new technologies.
  - Building new class A of healthcare facilities.
  - Lack of competitors nationally.

  Threats:
  - Some healthcare facilities will not seek for technologies.
  - Some healthcare facilities will raise the examination cost.
  - Some patients will be afraid of robotics to measure their vital signs.
  - Advancement in healthcare robotics abroad.

We did a survey to take the people's opinion about our project. We found great interactions from the people as we got 236 responses that proved the significance of the project. We asked many people with different educational background some questions and the results were good. We found that 68.2% believed that healthcare robots would replace the medical staff in the future. We also found that 89% thought that healthcare robot could measure your temperature. There were 77.5% would not be afraid if a robot would measure their temperature.

Chapter 3: Literature Review

Table 1. Literature review matrix [5] – [7]

Author / Origin / Date	Purpose	Target Population	Methodology	Analysis & Results	Conclusions	Future Scope
Mahmod El-komy Egypt 2020	Prevent the covid-19 transmission and limit the risk of infection.	Medical staff and patients.	A remote controlled robot is standing and moving on four wheels, which tested at a private hospital in Tanta, Egypt and integrated with IOT to send the data for anyone and anywhere.	A robot with a head and arms. The robot is able to take patient temperature, run a PCR test, perform x-rays scan.	The robot is much precise than humans in performing the medical staff tasks in the fight against coronavirus.	We will use the robot in crowded places such as banks and airports with oral communication between people and the robot.
DROID team China 2012 then used in Covid-19 periods	Prevent the covid-19 transmission and limit the risk of infection.	Anyone inside any healthcare facility.	A group of robots applied in a medical center in Kigali, Rwanda. Besides record vital signs, it can also remind people to wear masks and wash hands.	They are able to take a patient's temperature if the patient stand in front of an integrated thermal camera in the robot head or arm.	A group of robots reduces the burden on nurses and makes them concentrate in the most other important tasks.	It will be integrated with other sensors to make more tasks such as performing PCR tests
Esben Ostergard Denmark 2020	Make the covid-19 tests faster by taking swabs in a short time.	Nurses and patients.	University of Southern Denmark created a fully automated robot "say AAH" that depends on machine learning and	A robotic arm with a long swab can reach deep targets inside the patient's throat	A safe robotic arm conduct the test in a fast, safe, and precise manner.	They want to make the process faster than 7 minutes to target many patients in no time.

			computer vision to reach the spot inside the patient's throat.	and collect a sample then put it into a container to analyze it in no more than 7 minutes.		
Diligent robotics China 2017 and used in Covid-19 periods	Taking over time-consuming tasks so that the medical staff can focus on much vital responsibilities.	Nurses and physicians.	A robot called "Aimbot" with an integrated camera applies social distance by warning people and gives patients masks using the arms.	Repetitive tasks 24 hours. It can also deliver drugs and supplies from one room to another.	let medical staff spend much time in doing important tasks	We will depend on it in disinfection by spraying disinfectant.
Qihan robotics China 2016 and used in Covid-19 periods	Reducing the risk of catching covid-19 in the fight against the virus.	Medical workers	A robot called "Sanbot" is able to record patient data such as blood oxygen levels. We may attach microphone and camera to the robot.	Checking the workers' health day and night.	We use the design for offices to track the workers' health but then used in tracking infectious diseases like covid-19.	We will use it globally in hospitals, offices, and other places.
ZoraBots robotics Belgium Used in covid-19 periods	People interaction in the quarantine time	Corona virus infected people	An equipped robot with cameras, microphones, and a screen to make people virtually communicate easily.	Make people interact together at a time when things are mostly in hold.	It used in zoological park in Japan when it was closed due to the spread of coronavirus	We can use it anywhere and anytime to make interaction with each other easily.

Chapter 4: Materials and Methods

4.1 Material Selection

There were many materials to fabricate our prototype such as plastic, acrylic, aluminum and steel [8]. We selected depending on the pros and cons of each material. We chose plywood sheets for prototyping and we finalized our model with the same material. The next table shows the pros and cons of plywood sheets to serve our prototype.

Table 2. Pros and cons of prototyping material

Pros	Cons
• It has many types from thin to thick ones. • Very cheap. • No complex machines to cut or drill holes. • Wood is not waterproofed, but we can put a cheap material into it to waterproof it.	• It can be broken easily. • It damages with time more rapidly. • Although we can waterproof wood, it is not suitable for long exposure to water.

4.2 Prototype Fabrication

In the next few lines, we will introduce the fabrication process. We will also introduce the proof of concept and analysis of the final mechanical design. The fabrication process is how to choose the best material to build your model with the selected components.

4.2.1 Mechanical Calculations of the Prototyping Material

Initial design from plywood sheets to test motor capabilities and the mechanism performance. In the real life, the fabricated arm with motors and plywood material weighed 0.65 kilograms. We calculated the max load or force by dividing torque over distance; take the shoulder motor for example, as it was the most important motor due to carrying the whole armload. The torque of this motor was 30 kg.cm and the whole arm length was 60 cm. If we divided 30 kg.cm over 60 cm, we got 0.5 kg. This meant that the end effector of the whole arm could carry until 0.5 kg and it might reach 0.6 when overstressed. We could apply this formula at other arm links: $F = \frac{\tau}{l} \dots (1)$

Where: $\boldsymbol{F}$ (kg) is the max load at the link end effector.
$\boldsymbol{\tau}$ (kg.cm) is the motor torque.
$\boldsymbol{l}$ (cm) is the arm length.

Our material was plywood hinged over rotational supports or joints. Every four plywood sheets made a box or link. To calculate the weight of one sheet, we needed to know that plywood density was 680 kg/m^3. For example, we would calculate the upper link of 15 cm in length. We would use this equation and we could apply for other arm links: $\rho = \frac{m}{V} \dots (2)$

Where: $\boldsymbol{\rho}$ (kg/m^3) is the plywood density.
$\boldsymbol{m}$ (kg) is the sheet mass.
$\boldsymbol{V}$ (m^3) is the link volume.

Multiply length, width and height values, which were 0.15 m, 0.05 m, and 0.003 m respectively to get the volume result, which was 0.0000225 m^3. Then we got the mass by multiplying the volume into plywood density and the result was 0.015 kg.

4.3 Motor Selection and Hardware Components

4.3.1 Motors Selection

We mentioned above in section 4.2.1 the length and the weight of the material of each part of the arm, so we needed to calculate the torque of the motors as in equation. 1.

We have calculated the weight of each part in previous section, so we could calculate the torque of the motors.

To control arm movements, we had two choices to use servo or stepper motors, but we have chosen to use servomotors with embedded encoder over stepper motors to control arm movements because of large torque, low weight, cost, feedback of servomotor, as shown in the next table. We used four servomotors to control arm movements in four DOF.

Table 3: The comparison between servo and stepper motors [9]

	Servo motor		**Stepper motor**	
Model	FB5311M	FS6530M	NEMA 23	NEMA 34
Weight	64 g	137 g	700 g	8400 g
Speed	40 RPM	45 RPM	400 RPM	1500 RPM
Encoder	Internal magnetic encoder		No feedback encoder	
Operating voltage	6 V	6 V	3.2 V	3.8 V
Operating current	1.5 A	1.5 A	3 A	3 A
Cost	275 EGP	750 EGP	450 EGP	3500 EGP
Precision and accuracy	Accuracy of both is the same.			
Torque	15.5 Kg.cm	30 Kg.cm	9 Kg.cm	Kg.cm

4.3.2 Hardware Components

- **Arduino uno:** We used arduino to take the feedback of the motors and control them.
- **Arduino IDE:** We used this IDE because it is suitable for the arduino board; we applied our control system of the motor with it.
- **Encoder:** We used (potentiometer) as a feedback for the large servomotor to measure the angle and direction of rotation of the servomotor to know its current position.
- **Power supply:** We used (SMPS Input 220Vac / Output +12Vdc/20A) to power the microcontroller and servomotors.
- **Gearbox:** We used gearbox for largest servo to provide more torque to avoid the errors that could happen. (Gear ratio 2:1)

4.4 Kinematics

We can define kinematics as a fundamental and classic topic in robotics that studies the link between a robot's joint coordinates and its spatial layout. Kinematics can provide very exact computations in a variety of situations, such as putting a gripper in space. Designing a system to move a tool from point A to point B, or determining whether a robot's motion would collide with obstacles. Kinematics simply considers the immediate values of the robot's coordinates, ignoring the movement of the robot under forces and torques [10].

In manipulator robotics, there are two kinematic tasks [11]:

- **Direct (forward) kinematics**: The robot arm's joint relations (rotations and translations) are provided. Task: What are the end effector's orientation and position?
- **Inverse kinematics:** The required end-effector position and orientation are provided. What joint rotations and orientations are required to accomplish this?

Fig. 4 shows the difference between forward and inverse kinematics:

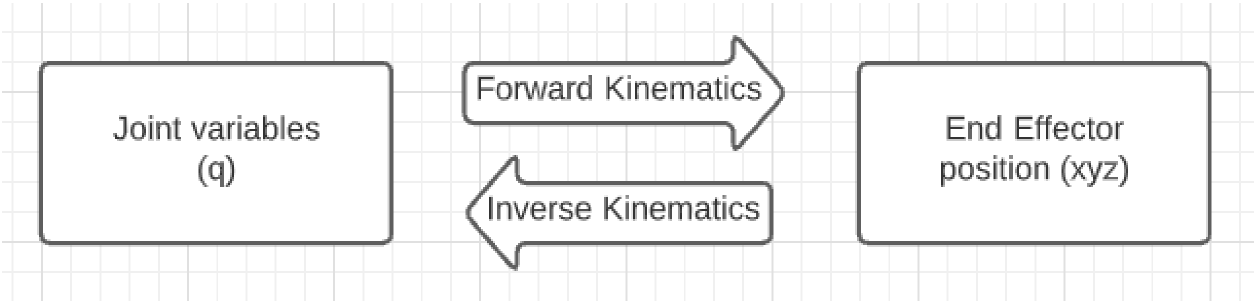

Figure 4: The difference between forward and inverse kinematics

4.4.1 Forward Kinematics

4.4.1.1 Used Tools

We used software for simulation like Matlab version R2016a and Robotics Toolbox for Matlab (release 9.10). We downloaded the Robotics Toolbox from the following link: https://petercorke.com/toolboxes/robotics-toolbox/

After downloading, we added this toolbox to Matlab by changing the current directory of Matlab to the folder of toolbox, then run startup_rvc.m, as shown in Fig. 5.

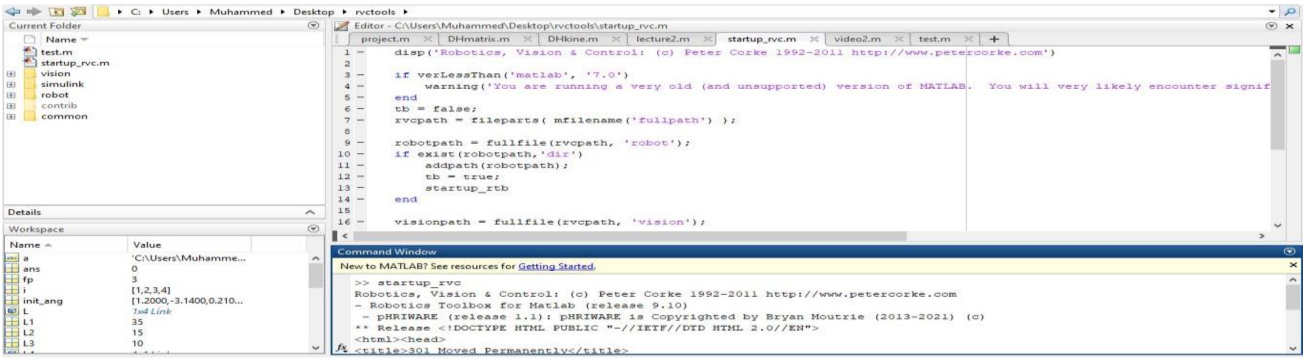

Figure 5: Starting setup

4.4.1.2 Forward Kinematics Steps

How to Calculate the Forward Kinematics of a Robot in five simple steps?

1. Get a pencil and paper:

This simple task forces you to consider the actual physical configuration of the robot carefully, avoiding incorrect assumptions that can cause problems later in the coding process. We can draw a kinematic chain in a variety of ways. Choose your preferred style [12].

2. Figure out your axes:

We must draw the axes onto each joint as the next step. Each moveable joint in the Denavit-Hartenberg (DH) method has its own axis. Working with the robot will be simple if you set up your axes appropriately. If you set things up incorrectly, you will have many hassles. The two important axes to work out are:

a) Z-axis — for a revolute joint, the z-axis should be on the axis of rotation, while for a prismatic joint, it should be on the axis of extension.
b) X-axis – The x-axis should run parallel to the "common normal," which is the shortest orthogonal line connecting the prior and current z-axes [12]. For drawing x-axis, there are three cases:
   I. New Z-axis is parallel to old Z-axis, so the new X-axis has the same direction of common normal to the next link and start from the origin of the new Z-axis for simplicity, as shown in Fig. 6 [13].
   II. New Z-axis intersects with old Z-axis, so the new X-axis has a direction perpendicular to the two axes from the intersection point to the next link, as shown in Fig. 6 [13].
   III. New Z-axis is not parallel and not intersected with the old Z-axis, so the new X-axis has the same direction of common normal of two axes to the next link, as in Fig. 6.

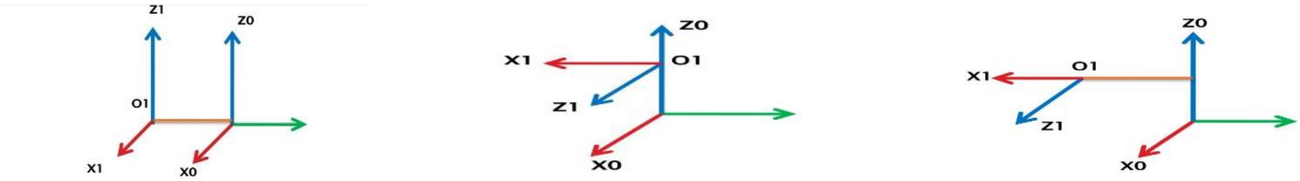

Figure 6: The cases of parallel axes, intersecting axes in one point, no parallel and no intersecting axes, respectively

3. <u>Remember your end effector:</u>

The purpose of computing forward kinematics is to be able to determine the end effector position based on the joint positions. We describe the end effector as a single distance from the last joint in most tutorials. For a simple "open-close" gripper, this is sufficient. However, because current grippers are much more complicated, it is worth thinking about how the end effector works [12].

4. <u>Calculate the DH parameters:</u>

The characteristics of DH are frequently necessary to load a robot model into a simulator and begin performing any type of analysis. The DH parameters divide each of the robot's joints into four parameters, each of which is measured in relation to the previous joint. They are determined using the "common normal" mentioned above. The common normal has a length of zero if the prior z-axis overlaps the present z-axis, which is typically the case. The parameters are [12]:

   I. **<u>d</u>** - The distance between the previous x-axis and the current x-axis, along the previous z-axis, as in Fig. 7.
   II. **<u>θ</u>** - The angle around the z-axis between the previous x-axis and the current x-axis, as in Fig. 7.
   III. **<u>a</u>** - The length of the common normal, which is the distance between the previous z-axis and the current z-axis, as shown in Fig. 7.
   IV. **<u>α</u>** - The angle around the common normal to between the previous z-axis and current z-axis, as shown in Fig. 7.

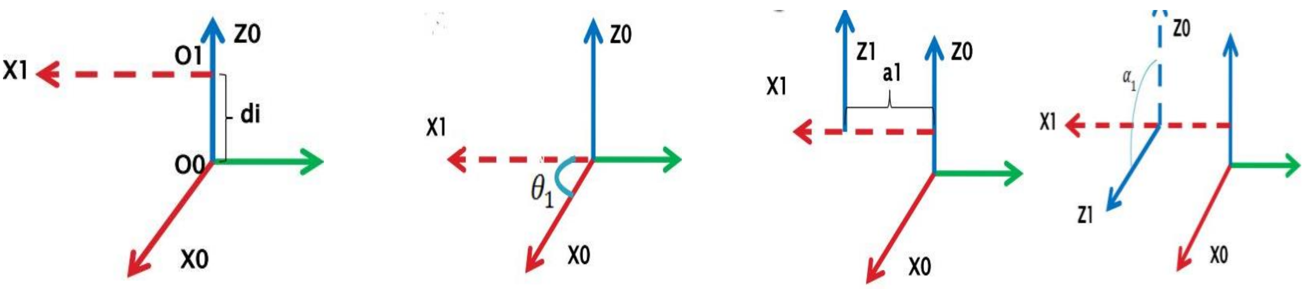

Figure 7: Illustration of d, θ, a, and α, respectively

5. <u>Create a table of link parameters like the next one and substitute these parameters to form the homogenous transform matrices A_i then form the transformation matrix (T) [13]:</u>

Table 4. The table of DH parameters for each link

Link	a	α	d	θ
1				
2				
3				
4				

The general homogenous transformation matrix as in Fig. 8, which relates to specific link and The final transformation matrix is: T = $A_1 * A_2 * A_3$ ………. * A_n

$$A_i = \begin{matrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{matrix}$$

Figure 8: The general transformation matrix

4.4.2 Inverse Kinematics

4.4.2.1 Introduction

Inverse kinematics is the mathematical process of calculating the variable joint parameters needed to place the end effector in the desired place [14]. The joint parameter in our robotic arm are the joint angles. We used iterative numerical method to calculate the joint angles given the desired position of the end effector. This method uses the Jacobean matrix (***J***) which is a linear approximation of a differentiable function near a given point as basis to calculate the joint parameters [15]. Fig. 9 shows an intuition of the Jacobean. Consider the notion $\boldsymbol{\theta}$ to be the joint variables and e to be the position of the end effector.

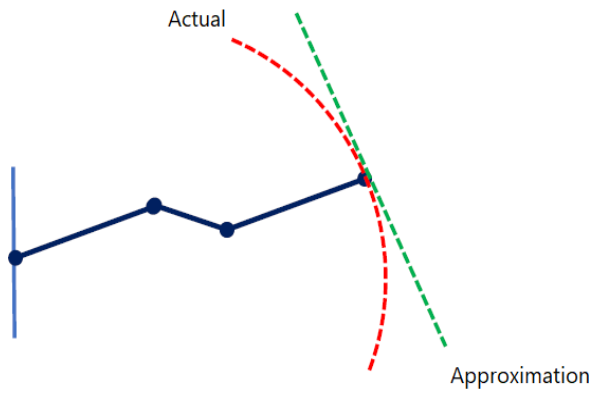

Figure 9: Jacobean intuition

$\theta = [\theta_1\ \theta_2 \quad \theta_M]$ … (1) & e = $[e_1\ e_2 \ldots e_N]$ … (2)

The Jacobean is matrix of a partial derivative of the entire system that defines how the end effector e changes relative to instantaneous changes in the system: $J = \frac{de}{d\theta}$ …. (3)

From (3) we can write: $de = J\ d\theta$ …. (4)

Where: $\theta = [\theta_1 \theta_2 \theta_M]^T$ & e = $[e_x\ e_y\ e_z]^T$

Fill in the Jacobean matrix: $J = \begin{bmatrix} \frac{\partial e_X}{\partial \theta_1} & \frac{\partial e_X}{\partial \theta_2} \cdots \frac{\partial e_X}{\partial \theta_M} \\ \frac{\partial e_Y}{\partial \theta_1} & \frac{\partial e_Y}{\partial \theta_2} \cdots \frac{\partial e_Y}{\partial \theta_M} \\ \frac{\partial e_Z}{\partial \theta_1} & \frac{\partial e_Z}{\partial \theta_2} \cdots \frac{\partial e_Z}{\partial \theta_M} \end{bmatrix}$ ... (5)

Our goal is to calculate the joint variable, so we pre-multiply (from the left) (4) by J^{-1} and this yields the following: $d\theta = J^{-1}$ de ... (6)

To calculate the Jacobean numerically, let's examine one column of the Jacobean matrix which is $\frac{\partial e}{\partial \theta_1} = \left[\frac{\partial e_X}{\partial \theta_1} \frac{\partial e_Y}{\partial \theta_1} \frac{\partial e_Z}{\partial \theta_1}\right]^T$, we can add $\Delta\theta$ to θ_i then we can calculate how the end effector moves:

$\Delta e = e' - e$ …. (7)

From (7), we can approximately write $\frac{\partial \mathrm{e}}{\partial \theta_1} = \left[\frac{\partial e_X}{\partial \theta_1} \frac{\partial e_Y}{\partial \theta_1} \frac{\partial e_z}{\partial \theta_1}\right]^T$ as following $\frac{\Delta \mathrm{e}}{\Delta \theta i} = \left[\frac{\Delta e_x}{\Delta \theta i} \frac{\Delta e_y}{\Delta \theta i} \frac{\Delta e_z}{\Delta \theta i}\right]^T$ and then fill in the Jacobian matrix the same way.

After we computed the Jacobean numerically, we calculated the inverse of the Jacobean using two methods: the pseudo inverse and Jacobean transpose.

4.4.2.2 Pseudo Inverse

From (4) we have: $de = J\, d\theta$

Multiply with J^T both sides: $J^T\,\mathrm{de} = J^T J\,\mathrm{d}\theta$ … (8)

Then multiply by $(J^T J)^{-1}$ both sides: $(J^T J)^{-1} J^T . de = (J^T J)^{-1}(J^T J). d\theta$ … (9)

$$(J^T J)^{-1} J^T . de = d\theta \; \ldots (10)$$

$$J^{+}. de = d\,\Delta\theta \; \ldots (11)$$

Where J^{+} is the pseudo inverse and equals to: $J^{+} = (J^T J)^{-1} J^T$ ... (12)

4.4.2.3 Jacobean Transpose

It is simply using the transpose of the Jacobean matrix as the inverse of the Jacobean: $J^{-1} = J^T$ ..(13)

4.4.2.4 Algorithm

We used the Algorithm of the Jacobean method for solving the inverse kinematics, which implies that the Jacobean can only be used as an approximation that is valid near the current configuration. Therefore, we must repeat the process of computing the Jacobean and then taking a small step towards the goal until we converge to the desired position.

The pseudo code for the algorithm is the following [16]:

While (e is too far from goal)

{ Compute the Jacobean matrix J.

Compute the pseudo inverse of the Jacobean matrix J^{+}.

Compute the change in joint DOFs $\Delta\theta = J^{+}.\Delta \mathrm{e}$

Apply the change to DOFs, move a small step of $\alpha\Delta\theta$: $\theta = \theta + \alpha\Delta\theta$

}

4.4.2.5 Matlab Code

We used Matlab version 9.4.0.813654 (R2018a) and Peter Corke robotic toolbox version RTB10.x. We used the function ikine(T, q0,m,options) which is used for under actuated robots (less than 6 DOF) . The first argument (T) represents the desired position that we want the end effector to be in, the second argument (q0) represent the initial values of the joint parameter, the third argument (m) represents mask vector (1x6) which specifies the Cartesian DOF that will be ignored in reaching a solution. The mask vector has six elements that correspond to translation in X, Y and Z, and rotation about X, Y and Z respectively. The value should be 0 (for ignore) or 1, the forth argument (options) represents a set of options to provide more control over the process of calculating the joint parameter which is presented in Table. 5.

Table 5: Options description

Option	**Description**
'pinv'	use pseudo-inverse instead of Jacobean transpose (default)
'ilimit', L	set the maximum iteration count (default 1000)
'tol', T	set the tolerance on error norm (default 1e-6)
'alpha', A	set step size gain (default 1)
'varstep'	enable variable step size if pinv is false

‘verbose’	show number of iterations for each point
‘verbose=2’	show state at each iteration
‘plot’	plot iteration state versus time

4.5 Bang-Bang Control

Bang-bang control is a sort of control system that switches things on or off mechanically or electrically when reaching a specified objective (set point). Bang–bang controllers, also known as two-step controllers, on-off controllers, or hysteresis controllers, are usable in a wide range of residential and industrial control systems.

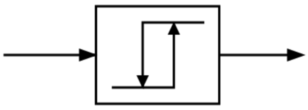

Figure 10: Bang-bang symbol

Our servos are continuous servos, when we give any motor a PWM it reaches to it and considers this position is the origin so it continues rotating with any PWM we give but it has differences in velocity. We used the embedded magnetic encoder of the servo to take feedback so we can detect its position. After detection of the position, we can hold the servo but the servo will rotate again. Here comes the Bang-bang control role. The motor holds the position temporary and if the position changes, the motor starts rotating in the opposite direction. This depends on the threshold we used in our implementation of the system.

Table 6: Thresholds of directions

Maximum PWM in CCW	PWM for Temporary Hold	Maximum PWM in CW
0	90	180

We set PWM of 10 as a tolerance (+ve or -ve). Fig. 11 is an example to explain how the control system works. Fig. 12 shows the control flow. We assume PWM 120 to move the motor. The motor will go to 120, we know this position from the embedded encoder then we give the motor 90 for temporary hold, when the position changes to another one that is out of range of our tolerance” position 140 for example” (after holding) the motor starts to rotate in CCW until it reaches 120.

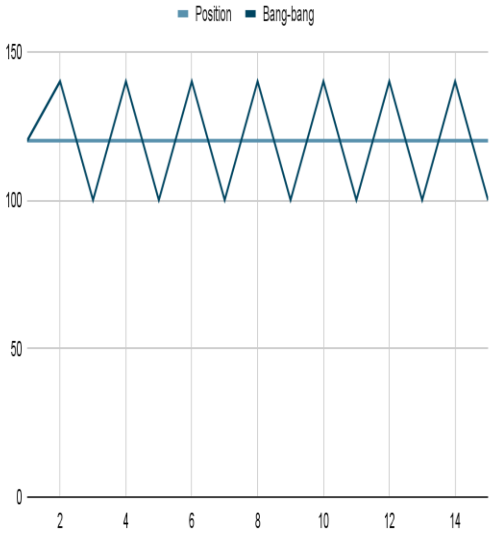

Figure 11: How bang-bang control works

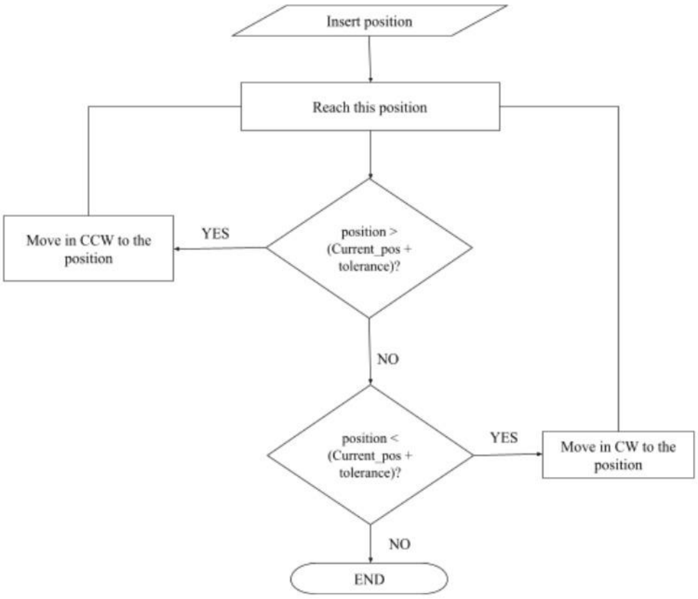

Figure 12: Flow of the control system

Chapter 5: Results

5.1 Prototyping Used Machines

We used two machines to build the final prototype as in Fig. 17 which are laser cutter (MORN MT-L1410), as shown in Fig. 13 to make the arm links and router simplex as shown in Fig. 14 to make the fixed body. We cut the sheets to make boxes of our desired design 2D mounted assembly. We cut every link separately then make a box contains every motor then joined with the next link.

Figure 13: Laser cutter machine

Figure 14: The router simplex

5.1.1 Dimensions

A humanoid right robotic arm with four DOF to move within limitations. It consists of four links with different lengths as shown in Fig. 15. We designed our model by a specific software, which is AutoCAD (version 2020) as in Fig. 16. The total arm length is 60 cm. These four motors are able to move or carry different weights and you should put into consideration that the shoulder motor is the most important one as it carries the whole armload.

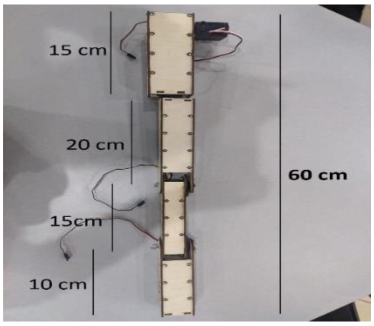

Figure 15: The links' lengths

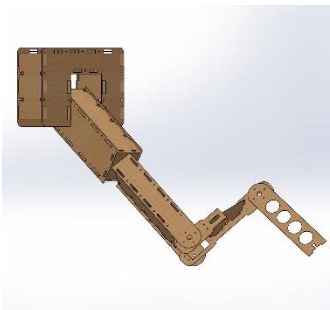

Figure 16: AutoCAD design

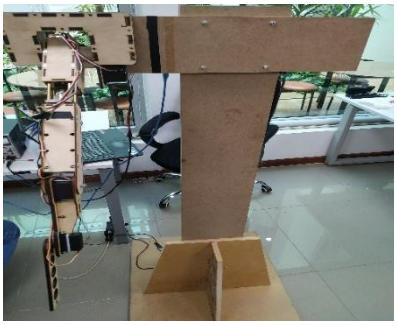

Figure 17: The final prototype

5.2 Kinematics

5.2.1 Forward Kinematics

After applying DH parameters, the next table shows the homogenous transformation matrix and the next figure shows the output after substituting with the values of different theta:

Table 7: The homogenous transformation matrix after applying DH parameters

Link	a_i	α_i	d_i	Θ_i
1	0	Pi/2	0	Θ_1
2	0	-Pi/2	L_2	Θ_2
3	L_3	0	0	Θ_3
4	0	Pi/2	0	Θ_4

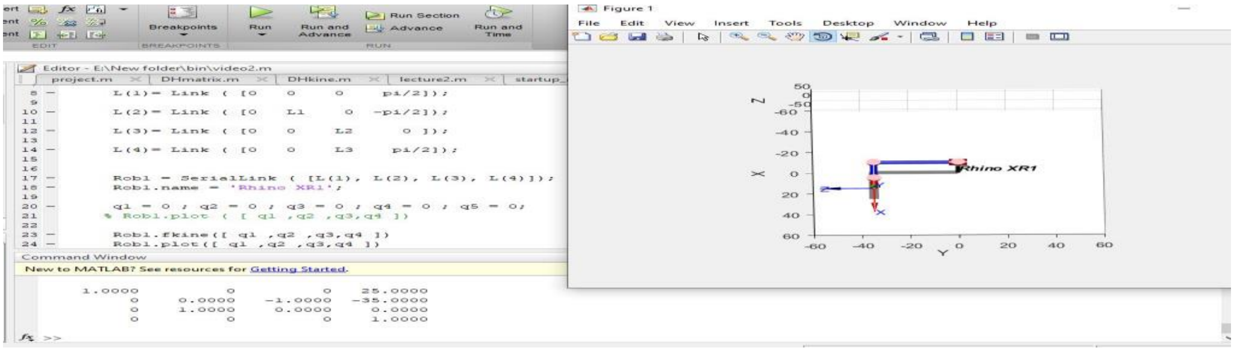

Figure 18: The output after substituting with θ_1=0, θ_2=0, θ_3=0, and θ_4=0

5.2.2 Inverse Kinematics

5.2.2.1 Jacobean Transpose and Pseudo Inverse

We made three experiments that shows how the end effector is reaching the given positions using the Jacobean transpose and pseudo inverse methods. All experiments done below is made with the default values of the function ikine() described in Table. 5.

- **Position (10, 50, 0):** Jacobean execution time: 0.430788 seconds & Pinv execution time: 0.434738 seconds.

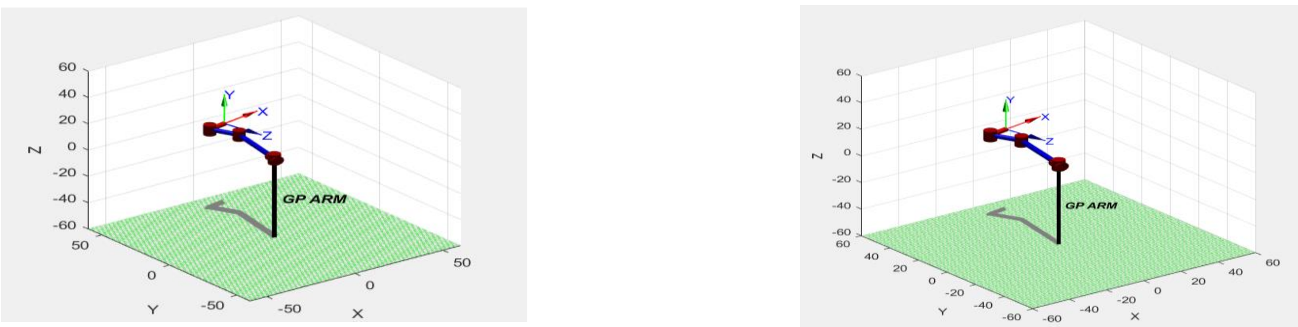

Figure 19: Robot configuration using Jacoean method (left) and pseudo inverse (right)

Figure 20: plot iteration state, Jacobean method (left) and pseudo inverse (right) where q: solution of joint angels in radians, dq: change in joint angels, and e: error

- **Position (15, 30, 20):** Jacobean execution time: 0.550430 seconds & Pinv execution time: 0.759835 seconds.

Figure 21: Robot configuration using Jacobean method (left) and pseudo inverse (right)

Figure 22: plot iteration state, Jacobean method (left) and pseudo inverse (right) where q: solution of joint angels in radian, dq: change in joint angels, and e: error

- **Position (-20, 25, 10):** Jacobean execution time: 0.309141 seconds & Pinv execution time: 0.430384 seconds.

Figure 23: Robot configuration using Jacobean method (left) and pseudo inverse (right)

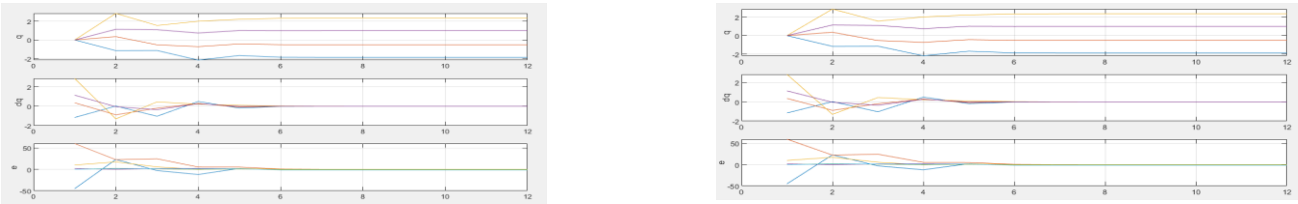

Figure 24: plot iteration state, Jacobean method (left) and pseudo inverse (right) where q: solution of joint angels in radians, dq: change in joint angels and e: error

5.2.2.2 Initial Values for Joint Angles

We studied the effect of choosing different initial values for joint angels on the Arm configuration and the result were as follows:

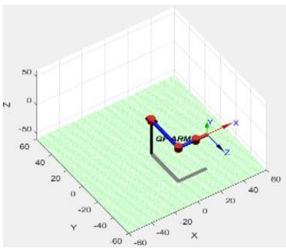

Figure 25: initial [0 0 0 0] at position (25,-35,0)

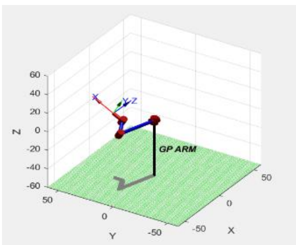

Figure 26: initial [0 0 0 0] at position (-20, 25, 10)

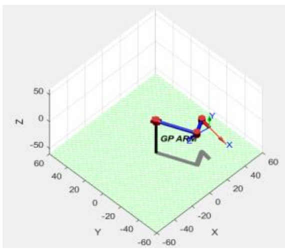

Figure 27: initial [0.5 0.6 1 2] at position (25, -35, 0)

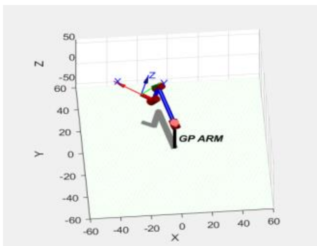

Figure 28: initial [10 0.5 20 30] at position (-20, 25, 10)

5.3 Validation of Motors

Our servomotors provide high torque with high speed but the accuracy of our robotic arm would not be accepted. Therefore, we used trial and error method to provide the best torque that achieves best accuracy. For the largest motor, we used the gearbox to get better control for the arm because this motor would not achieve the wanted torque as shown in Fig. 29. For the second DOF we used flexible shaft coupling to provide better holding for the rest of the arm as shown in Fig. 30.

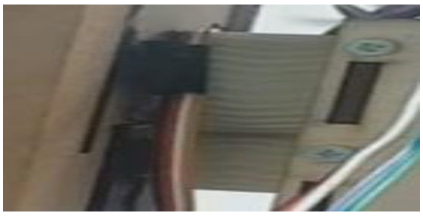

Figure 29: 3D printed gearbox

Figure 30: Flexible shaft coupling

5.4 Validation of Microcontroller and Power Supply

As we before in section 4.3.2 , we used arduino uno to control the motors. Arduino has 8 bits resolution that gives us a narrower range of feedback signals coming from the servo encoder. It was accepted in our project but sometimes it made motors consume more time to hold at its accurate position. For the power supply, it provides 12v with more than enough current for our system, but the motors consume 6v. Therefore, we used a step down converter to get this 6v as shown in Fig. 31.

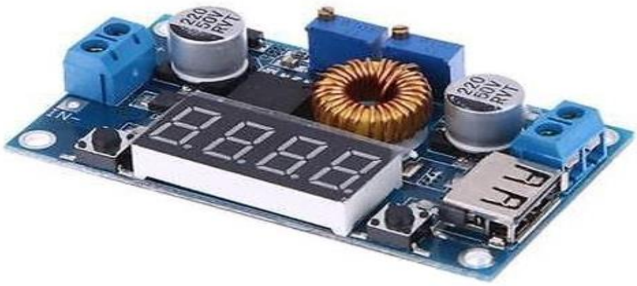

Figure 31: Step down converter

5.5 Validation of Bang-Bang Control

We used Bang-bang in our system. It depends on feedback from the encoder. Bang-bang control provided good accuracy in our application. Bang-bang control was easy to implement compared to PID control and consumed less execution time. We used the feedback to check the position of the motor and hold it as shown in Fig. 32. We tested the system with external force. After the arm reached a certain position we changed this position with external force, the bang-bang control checked the position and moved the arm to its old position. The limitation of this control method was that the motor sometimes consumed more time to identify the position and that made the motor rotating in CCW and CW to stop. In addition, the bang-bang control needed a microcontroller with high resolution to give a wider range of readings of the encoder.

Figure 32: The arm reached the position

Chapter 6: Discussion

6.1 Material Development

Our arm consists of some links and joints. There will be sensor at the end effector in x-y plane, so we wanted the arm to work as a manipulator. It must be rigid. The links should avoid tangency and resist bending and breaking.

Plywood or we can call it rigidity links with joints is the best mimic for a humanoid arm and gives us stability. It was the easiest manufacturing in prototyping. We chose plywood to test functionality and for rapid prototyping.

However, for reliability, we could use other materials such as the previous mentioned materials as plywood had some drawbacks.

The other materials drawbacks were the cost. Some of them needed to complex machines. Some materials were difficult to find them. Heaviness was one of the drawbacks [5].

6.2 Kinematics

6.2.1 Forward Kinematics

We applied the concept of DH parameters in our robot and substituted with the values of {a, α, d, Θ} for each link as shown before and then put the table in Matlab to get the final transformation matrix (T) and to simulate the robot's position.

There are some limitations. The DH method is the most widely used technique for Forward Kinematics; however, it is not without flaws. One of its flaws is that it does not do a good job of handling parallel z-axes. Screw Theory representations, Hayati-Roberts, and other geometric modeling's are some of the possibilities. These may (or may not) be good approaches. However, most kinematic libraries do accept the DH parameters and for that reason, it is a reasonable approach, to begin with [12].

6.2.2 Inverse Kinematics

6.2.2.1 The Pros and Cons of Numerical Solution

The problem with numerical solution is that we must choose a good initial value for joints variables, we cannot guarantee a particular configuration as it differs according to the choice of the initial values. Finally, the numerical solution can be computationally expensive. On the other hand, the numerical solution provide a good approximation to the desired solution without the tedious process compare to the analytical process. It is also much feasible than the analytical solution especially in complicated configurations [15] – [17].

6.2.2.2 Initial Values

In the function ikine() We found out that the solution is sensitive to choice of initial values and the robot type, and a trial and error process is required to choose the best values as well as to find the limit , tolerance and step size. For example, the choice of zero initial value is a poor choice for some robots because it corresponds to kinematic Singularity [18]. However, in our case the best initial values were the zeros.

6.2.2.3 The Difference between the Jacobean Transpose and Pseudo Inverse

The Jacobean transpose method is faster than the pseudo inverse method as the result showed. Regarding the quality, there was no deference between the results produced by the two methods. However, it is safe to say that the quality of Jacobean transpose method is not good compared to the pseudo inverse method. Because the Jacobean is already an approximation to the solution and in this method, we do more approximation (a huge one) by taking the transpose of the Jacobean as the inverse of the Jacobean, which leads to a low quality. On the other hand, pseudo inverse method is slower but the quality is better [16].

6.2.2.4 The Pros and Cons of the Used Function

Joint limits are not considered in this solution, this approach allows a solution to be obtained at a singularity, but the joint angles within the null space are arbitrarily assigned [18].

6.3 Trajectory Planning

For robotic applications and automation in general, trajectory planning is a critical topic. The capacity to generate trajectories with specific features is critical for achieving considerable outcomes in terms of motion quality and ease of execution, especially at the high operating speeds required in many applications [19].

The purpose of trajectory planning is to generate reference inputs for the manipulator control system so that the intended movement can be accomplished. It can be assumed that a trajectory planning algorithm takes as inputs the geometric path, the kinematic and dynamic constraints of the manipulator; the output is the trajectory of the joints, or of the end-effector, expressed as a sequence of values of position, velocity and acceleration [19].

Simulating the trajectory planning in Matlab as shown in below figures are:

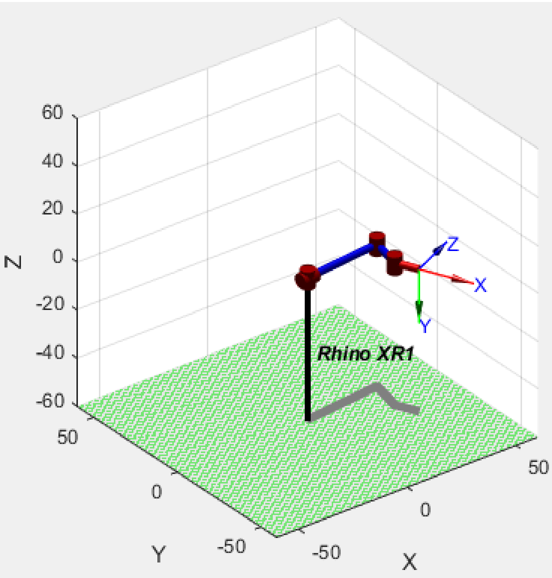

Figure 33: Initial position

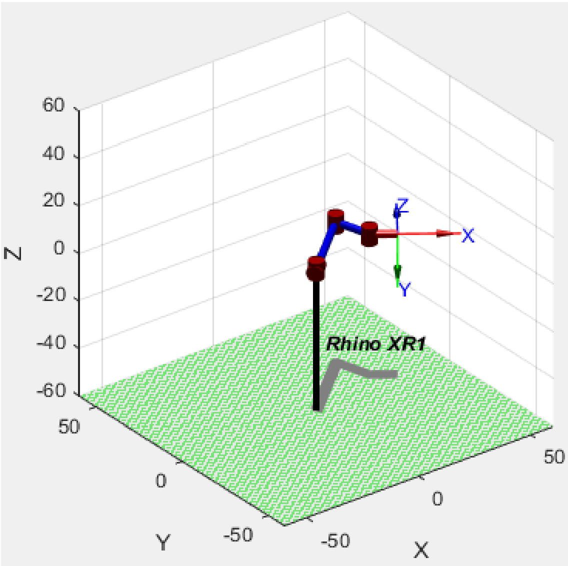

Figure 34: Second position

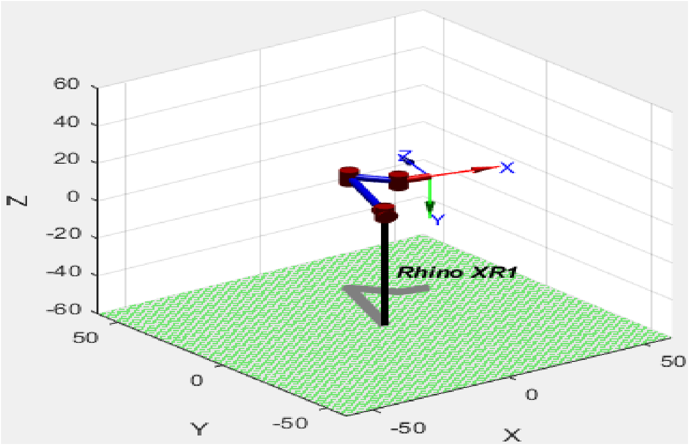

Figure 35: Target position

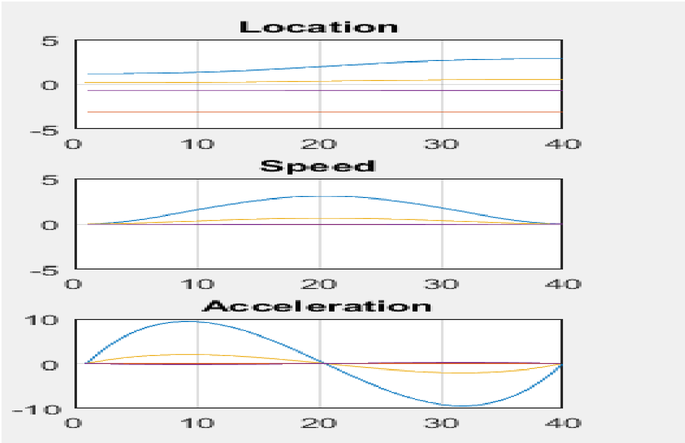

Figure 36: Location, speed, and acceleration

6.4 Components and Control System

We made many phases in our project. Therefore, we faced problems. In this section, we are going to describe them.

For the second DOF, we were using a 3D printed coupling for the motor but it crashed. Therefore, we used the flexible shaft coupling mentioned in the material section.

As mentioned section 5.5, to apply good bang-bang control we need a high-resolution microcontroller. At first, we used STM32F104 as shown in Fig. 37, it has a resolution of 12 bit. Therefore, we had wider range to our control and higher response with less time to reach the position. However, we had a problem with the communication protocol with Matlab inverse kinematics. We used arduino because the Matlab contains a library to apply the serial communication between it and arduino.

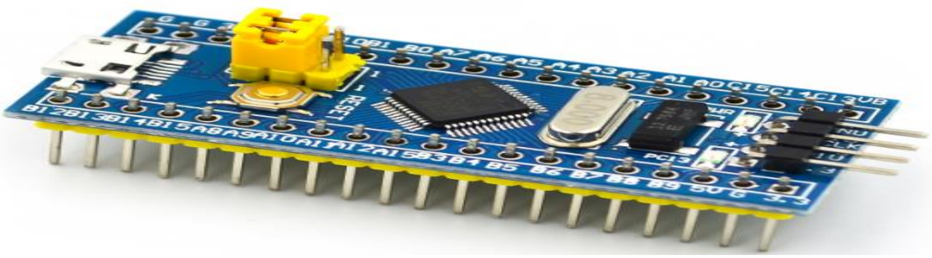

Figure 37: ARM STM32 Minimum System Development Board

To save your effort and time you need to make sure that all material will be compatible with each other.

We connected the servomotors to a series circuit. The motors work with 6V, the voltage reached 12V from an unstable power supply. We lost the largest motor because it got the largest voltage over the others.

In bang-bang implementation, we needed delays to move the arm and then check if the arm reached the wanted position. Nevertheless, these delays are block functions that make the microcontroller do nothing during each delay. It would be better if we used the timer concept so we could avoid this blocking to achieve a better real time application.

Chapter 7: Conclusion and Future Work

7.1 Conclusion

In summary, we built a fully controlled prototype for a humanoid robotic arm to assist the medical staff. We introduced a robotic system that will help all healthcare workers who are exerting their best in the fighting against the outbreak of Covid-19 crisis in different medical institutions by protecting them from catching this virus infection, so they can perform their tasks safely and quickly. Providing a relative cheap product for all medical institutions to face the coronavirus pandemic.

7.2 Future Work

We can enhance our prototype with professional improvements. We can equip the end effector of the whole arm with some sensors to measure vital signs such as temperature sensors and heart rate sensors. We can build the left arm and equip it with other sensors for example to detect blood oxygen level. We may integrate the arm with a robotic body to be autonomous and to use it in deliver drugs or supplies. We can also control the arm with any mobile application. We may equip the arm with visual communications.

References

[1]. Egypt: WHO Coronavirus Disease (COVID-19) Dashboard With Vaccination Data", Covid19.who.int, 2021. [Online]. Available: https://covid19.who.int/region/emro/country/eg.

[2]. H. Erdem and D. Lucey, "Healthcare worker infections and deaths due to COVID-19: A survey from 37 nations and a call for WHO to post national data on their website", International Journal of Infectious Diseases, vol. 102, pp. 239-241, 2021. Available: 10.1016/j.ijid.2020.10.064.

[3]. "The Role of Robots in COVID-19 Era", Swinburne University, Sarawak, Malaysia, 2021. [Online]. Available: https://www.swinburne.edu.my/campus-beyond/role-robots-covid-19-era.php.

[4]. "Full Page Reload", IEEE Spectrum: Technology, Engineering, and Science News, 2021. [Online]. Available: https://spectrum.ieee.org/robotics/medical-robots/how-robots-became-essential-workers-in-the-covid19-response.

[5]. X. Wang and L. Wang, "A literature survey of the robotic technologies during the COVID-19 pandemic", Journal of Manufacturing Systems, 2021. Available: 10.1016/j.jmsy.2021.02.005.

[6]. "Full Page Reload", IEEE Spectrum: Technology, Engineering, and Science News, 2021. [Online]. Available: https://spectrum.ieee.org/robotics/medical-robots/how-robots-became-essential-workers-in-the-covid19-response.

[7]. R. Staff, "Egyptian inventor trials robot that can test for COVID-19", U.S., 2021. [Online]. Available: https://www.reuters.com/article/health-coronavirus-egypt-robot-idUSKBN2852F6.

[8]. K. Matthews, "Materials to evaluate for designing and building robust robots", The Robot Report, 2021. [Online]. Available: https://www.therobotreport.com/materials-rugged-robot-design-building/.

[6]. "Full Page Reload", IEEE Spectrum: Technology, Engineering, and Science News, 2021. [Online]. Available: https://spectrum.ieee.org/robotics/medical-robots/how-robots-became-essential-workers-in-the-covid19-response.

[7]. R. Staff, "Egyptian inventor trials robot that can test for COVID-19", U.S., 2021. [Online]. Available: https://www.reuters.com/article/health-coronavirus-egypt-robot-idUSKBN2852F6.

[8]. K. Matthews, "Materials to evaluate for designing and building robust robots", The Robot Report, 2021. [Online]. Available: https://www.therobotreport.com/materials-rugged-robot-design-building/.

[9] "Servo Motors Qn 4: Which is better accuracy: Stepping motor or Servo motor?", *Orientalmotor.com.sg*, 2021. [Online]. Available: https://www.orientalmotor.com.sg/qa_det/qa_svmotors04/.
workers-in-the-covid19-response.

[10]. "Kinematics", Motion.cs.illinois.edu, 2021. [Online]. Available: http://motion.cs.illinois.edu/RoboticSystems/Kinematics.html.

[11]. Scribd.com, 2021. [Online]. Available: https://www.scribd.com/document/379670138/11KinematicsRobot-pdf.

[12]. A. Owen-Hill, "How to Calculate a Robot's Forward Kinematics in 5 Easy Steps", Blog.robotiq.com, 2021. [Online]. Available: https://blog.robotiq.com/how-to-calculate-a-robots-forward-kinematics-in-5-easy-steps.

[13] A. Zamel, Rebotics_07_Forward Kinematics Example 01 (DH parameters). 2017

[14] "Inverse kinematics - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Inverse_kinematics.

[15] Diva-portal.org, 2021. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1018821/FULLTEXT01.pdf.

[16] Cs.cmu.edu, 2021. [Online]. Available: http://www.cs.cmu.edu/~15464-s13/lectures/lecture6/IK.pdf.

[17] 2021. [Online]. Available: https://robotacademy.net.au/lesson/numerical-inverse-kinematics/.

[18] "SerialLink", Petercorke.com, 2021. [Online]. Available: https://www.petercorke.com/RTB/r9/html/SerialLink.html.

[19] A. Gasparetto, P. Boscariol, A. Lanzutti and R. Vidoni, "Trajectory Planning in Robotics", Mathematics in Computer Science, vol. 6, no. 3, pp. 269-279, 2012. Available: 10.1007/s11786-012-0123-8.