Design, control and implementation of SCARA robot for sorting missions with machine vision.

By

Ahmad Manasra　　Ashraf AL Sharif

Ala Salaymeh　　Yousef Natsheh

Supervisor: Dr. Yousef Sweiti

Submitted to the College of Engineering

in partial fulfillment of the requirements for the

Bachelor degree in Mechatronics Engineering

Palestine Polytechnic University

May 2018

Hebron – Palestine

Design, control and implementation of SCARA robot for sorting missions with machine vision.

Project Team

Ahmad Manasra

Ala Salaymeh

Ashraf Al Sharif

Yousef Natsheh

Submitted to the College of Engineering in partial fulfilment of the requirements for Bachelor degree in Mechatronics Engineering

Supervisor Signature

Dr. Yousef Sweiti

Testing Committee Signature

Dr. Jasem Tamimi

Tamimi

Dr. Hashem Tamime

Chair of the Departure Signature

....

May 2018

الإهداء

إلى مُعلمنا وقائدنا وشفيعنا وقدوتنا سيدنا مُحمد " صل الله عليه وسلم "

إلى بوصلة المسلمين ، الى عاصمة دولة فلسطين ، الى مدينة القدس المباركة

إلى من رسموا بدمائهم خارطة الوطن وطريق المستقبل وهندسوا بأجسادهم معاقل العزة والكرامة وإلى من هم أكرم منا جميعاً شهداء الوطن الحبيب

إلى الذين عشقوا الحرية التي تفوح منها رائحة الياسمين وتواروا خلف القضبان ليفسحوا لنا النور.. أسرانا البواسل

إلى من كلّت أنامله ليقدم لنا لحظة سعادة، إلى الذي طلبتُ منه نجمة، فعاد حاملاً السماء،إلى القلب الكبير أبي الحبيب

إلى التي رآني قلبها قبل عينيها وحضنتني أحشاءها قبل يديها ، إلى من كرّست حياتها لنجاحنا أُمي الحنونة

إلى من شاركونا ذكريات الطفولة وأحلام الشباب إلى أخواني الاعزاء

إلى كل من علمني حرفاً وغرس معاني التميز والابداع فينا الى معلمينا الاكارم

إلى من ضاقت السطور لذكرهم فوسعتهم قلوبنا..أصدقائنا الأعزاء

إلى من رسم معنا خطوات هذا النجاح ،الى من بذل جهده ووقته وكان لنا مرشداً وناصحاً وأخاً الى مشرفنا الحبيب الدكتور يوسف السويطي.

إليكم نهدي هذا العمل المتواضع

شكر وتقدير

"كن عالماً .. فإن لم تستطع فكن متعلماً ، فإن لم تستطع فأحب العلماء ،فإن لم تستطع فلا تبغضهم"

(الحمد لله الذي هدانا لهذا وما كنا لنهتدي لولا ان هدانا الله)

نحمد الله سبحانه وتعالى الذي سخر لنا قدره وجعلنا من المتعلمين ،فبعد مسيرة خمسة أعوام من رحلة بحث وجهد واجتهاد تكللت بإنجاز هذا المشروع ، نحمد الله عز وجل على نعمه التي منّ بها علينا فهو العلي القدير ، كما لا يسعنا إلا أن نخص بأسمى عبارات الشكر والتقدير الدكتور يوسف سويطي لما قدّمه لنا من جهد ونصح ومعرفة طيلة انجاز هذا المشروع . كما نتقدم بالشكر الجزيل لكل من أسهم في تقديم يد العون لإنجاز هذا المشروع، كما لا ننسى أن نتقدم بأرقى و أثمن عبارات الشكر والعرفان إلى جامعة بوليتكنك فلسطين وعمادة البحث العلمي لدعمهم لانجاز هذا المشروع، ونشكر ايضاً الاستاذ عبد الكريم المحتسب، الاستاذ مكاوي حريز والاستاذ علاء التميمي على الدعم الذي قدموه .

فلهم جميعاً منا كل الشكر والتقدير.

Abstract

This project aims to design, control and implement a four degree of freedom (4DOF) SCARA Robot to sort objects by their colors with taking an advantage of machine vision technology.

The first stage of the project after reviewing the previous works on SCARA robot and its application is to design a manipulator for this project. Then the mechanical structure of the manipulator will be modeled.

The machine vision system will determine the pose of each object with respect to the manipulator base frame. Based on these locations the path planner will generate the trajectory that the manipulator has to track. The Robot is expected to sort objects based on their color with precise, accurate repeatable picking and placing missions.

Keywords: SCARA, machine vision, Lagrange formulation, image processing, robotics control.

الملخص

يهدف هذا المشروع الى تصميم وتحكم وتنفيذ روبوت السكارا رباعي درجات الحرية حتى يتمكن من تصنيف الاجسام بناء على ألوانها بإستخدام انظمة الرؤيا الذكية. بعد مراجعة ودراسة هذا الروبوت والاطلاع على التطبيقات الخاصة به ، بدأت المرحلة الاولى للمشروع حيث تم تصميم ذراع الروبوت الصناعية ميكانيكياً.

تم تمثيل الجسم الميكانيكي للمشروع عن طريق المعادلات التفاضلية التي تصف ديناميكية حركة الذراع الالي لكي يتسنى تصميم المتحكمات التي سيتم من خلالها التحكم بالحركة .

ومن خلال انظمة الرؤية الذكية سيتم تحديد موقع كل جسم موجود في منطقة العمل وذلك بالاستناد الى نقطة مرجعية، وبالاستناد الى مواقع الاجسام سيقوم الحاسب الالي بتوليد خطة الحركة مع الزمن التي ستمثل المرجع التي يتوجب على المتحكمات تتبعها.

ما هو متوقع من الروبوت القيام به هو تصنيف الاجسام الموجودة على سطح العمل بناء على ألوانها بدقة مع تكرارية مهمة التقاط ووضع الاجسام في مكانها الصحيح.

Contents

List of tables

List of figures

1

Chapter 1

Introduction

1.1 Concept of the project

Selective Compliance Assembly Robot Arm or SCARA robot was invented by Hiroshi Makino in 1978 is shown in Figure 1.1 Its arm is rigid in the Z-axis and pliable in the XY-axes, which allowed it to adapt to holes in the XY-axes [1]. This was and still a revolutionary robot in many industries because of its speed and simple design.

In many industrial applications that require fast picking and placing operations, there is a need for such a robot because the human workers cannot handle these "boring" missions. SCARA robot is one of the options to do these missions accurately, fast and time efficient with less cost than other types of robots.

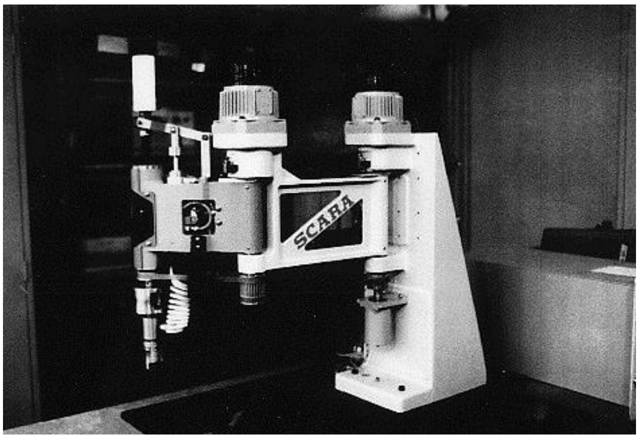


Figure 1.1: SCARA robot prototype made by Hiroshi Makino.

1.2 Recognition of the need

Some operations need to sort objects by their color, shape or both faster than human labors, therefore there is a need to construct a high speed and accurate robotic arm to do such missions. Some of these sorting missions will be done on the horizontal plane, so that it is reasonable to construct a simple robotic structure to do these missions hence there is no need here to construct a complex robotic arm like German Aerospace Center (DLR) manipulator or anthropomorphic robotic arm, SCARA robotic structure is a good choice in this case.

On the other hand, some operations are strictly needed to be done by robotics systems like sensitive and small-scale assembly operations such as in PCB assembly lines. Also, some sorting missions may be needed to be done in dangerous environments.

1.3 Project scope

This project aims to design, control and implement a 4DOF SCARA Robot to collect objects laied on horizontal plane and sort them into baskets by their colors with taking an advantage of image processing with computer vision within an $80 * 70 * 10\ cm$ working space.

The first stage of the project after determining the project limitations and assumptions is to design the manipulator. Then the mechanical structure of the manipulator will be modeled for the sake of controller design and simulation, more than one control algorithm will be used to control the manipulator to choose the most fitting controller for such a mission. The machine vision system will determine the pose of each object with respect to a reference frame, based on these poses the path planner will generate the trajectory that the manipulator has to track. The Robot is expected to sort objects based on their color with precise, accurate repeatable picking and placing missions.

1.4 Project objectives

- Optimize the mechanical structure of the robot to reduce the actuators energy consumption by setting the motors positions as close as possible to the base of the robot, in other words reduce the structure inertias.
- To simplify the poses sensing operation by taking advantage of computer vision system.
- Build a robust and reliable SCARA robot.
- To build an intelligent automated sorting system.

1.5 Methodology

The methodology to make this project contains several stages, theoretical calculations, which contains the dynamic model and design the mechanical structure for the robot.

After the theoretical calculations, several conceptual designs will be considered to show the connections between several components of the project (computer, interfacing circuits, controller, drivers and motors) and then choose the best design.

After that, the mechanical structure will be modeled in SOLIDWORKS software [2] to check the maximum deflection that occurs due to the maximum static and dynamic forces, after that, MATLAB and Simulink software [3] will be used to simulate its motion and its behavior according to the dynamic model and the desired controller.

1.6 Literature review

This section provides an overview of previous efforts of the industrial robot and construct selective compliance assembly robot arm " SCARA Robot".

1.6.1 The UNIMATE model ,1979

In 1961 - The first industrial robot was online in a General Motors automobile factory in New Jersey it was called UNIMATE, it used to pick and place and in assembly lines Figure 1.2. However, as computer development began, the developers were starting using the computer to control the robots.

In 1979 the researchers developed the UNIMATE, it has used image processing with a special camera called TV camera, it takes the picture between 10 to 66 milliseconds where the required as the research is between 100 to 500 milliseconds, so the image processed slower. The TV camera moved with respect to reference frame to detect the pose of the object. Also, it used for visual feedback to control the motion of a robot, this method is called visual servoing.

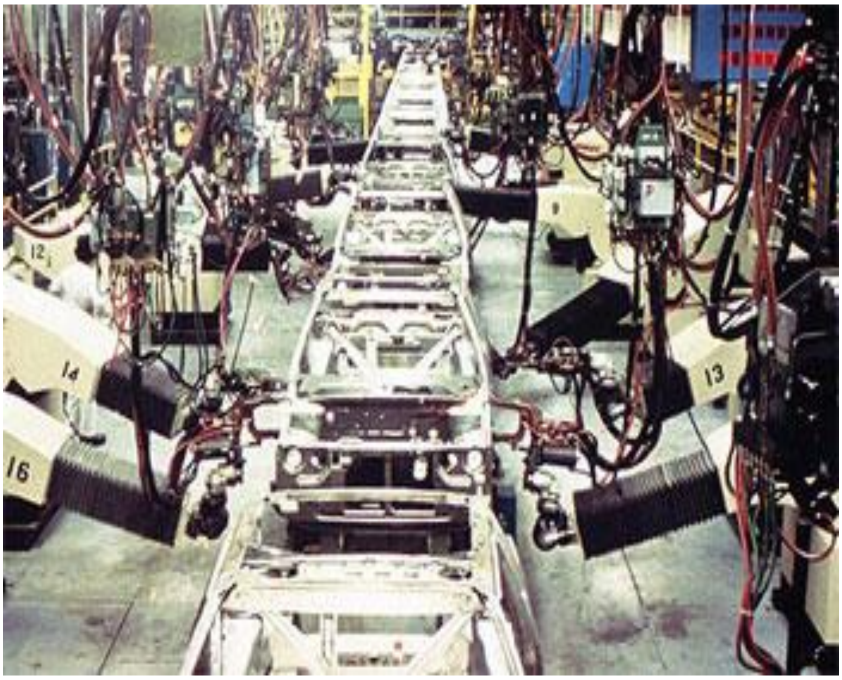


Figure 1.2: UNIMATE in assembly lines [4].

The robot is actuated hydraulically until the actual reading of the joint encoders agree with the ordered set points, it was inserted by the program that runs in the LSI-11 microcomputer. UNIMATE is collect between two system power: electric and hydraulic power.

1.6.2 SCARA robot first model ,1978

Selective Compliance Assembly Robot Arm SCARA was invented by Professor Hiroshi Makino form University of Yamanashi, Japan in 1978 [5]. The first prototype of the SCARA robot by Hiroshi M. shown in Figure 1.3. It has 4 DOF revolute, revolute, prismatic and revolute joints.

The main idea of SCARA robots that is it has a simple structure with high speed, accurate and precise manipulation in the plane by taking advantage of the third prismatic joint for pick and place missions.

The first industrial application of the SCARA robot was in assembling printed circuit board (PCB) of audio amplifiers, where eight robots were used in this assembly line.

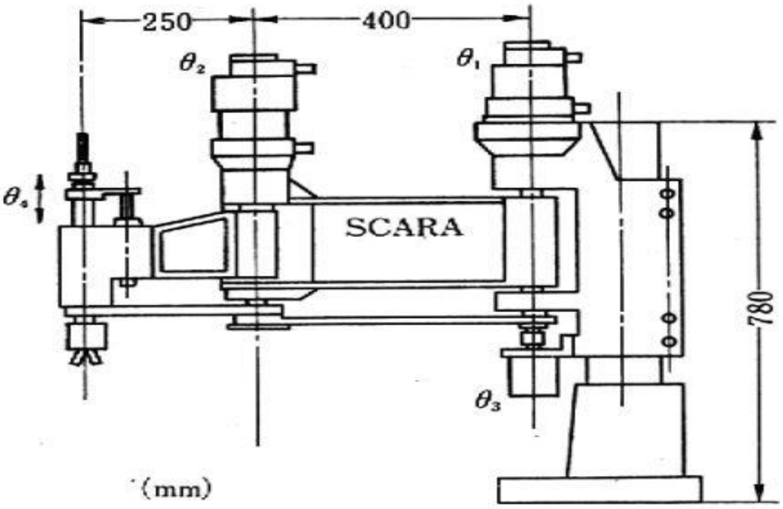


Figure 1.3: first SCARA detailed.

1.6.3 The SCARA Robot model, 2009

The researchers developed the SCARA robot to become more accurate and consumed low power by optimizing the mechanical structure for the manipulator. The 4-DOF SCARA shown in Figure 1.4, that is used for drilling task will be designed and developed using solid dynamic program and simulated using MATLAB and Simulink. Forward and inverse kinematics were derived using Denavit-Hartenberg notation [6].

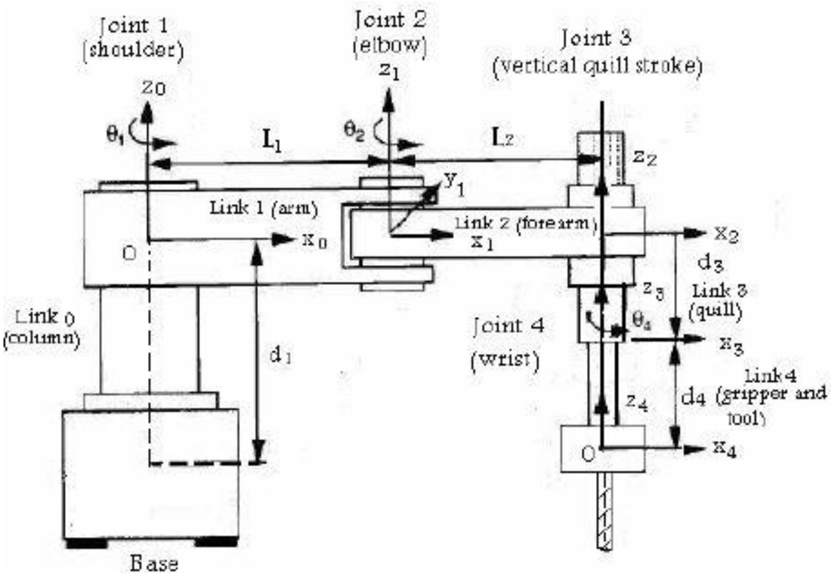


Figure 1.4: D-H Parameters for four- joint SCARA Robot.

The robot actuated using three electrical 24 V DC Servo motors and PD controller connect with SD program and MATLAB at real time.

1.6.4 The SCARA robot by resonance model, 2012

This model proposes an energy efficient method for pick and place tasks of SCARA robots. In the proposed method, an adaptive elastic device at each joint is effectively utilized to reduce the total energy of pick and place tasks. For practical pick and place tasks, start/end points must be changed depending on the required task [7].

The robot with mechanical elastic elements can generate high energy efficient periodic motions by changing kinetic energy into potential energy stored in the elastic elements.

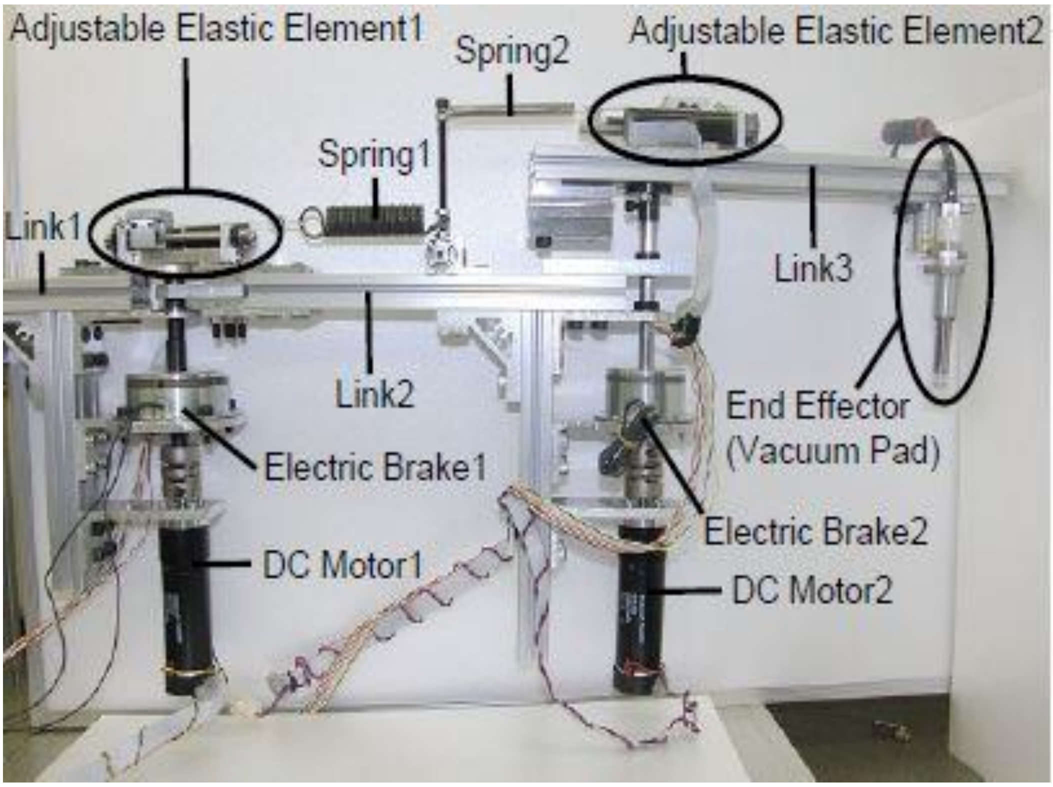


Figure 1.5: Front view of SCARA Robot.

This prototype shown in the Figure 1.6. The main component is electrical DC servo motor with gearbox with ratio (1:4.8), electric brake, adjustable elastic element (spring) and end effector (vacuum pad).

The prototype is used two power system: electrical system to actuate the Links (1, 2 and 3) and pneumatic system to actuate gripper (end-effector).

1.7 Time table

Time Table 1.1 shows the project reviewing, modeling and designing, work is divided in fifteen weeks in the first semester as following.

Table 1.1: Time table for the first semester.

Weeks Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Identification of Project Idea	🟦	🟦	🟦												
project requirement and collecting data			🟦	🟦											
Introduction Chapter 1				🟦	🟦	🟦	🟦								
Mathematical model Chapter 2				🟦	🟦	🟦									
Design Chapter 3					🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦			
Vision Chapter 4									🟦	🟦	🟦	🟦	🟦	🟦	🟦
Control Design Chapter 5												🟦	🟦	🟦	🟦

Time Table 1.2 shows the project implementation, work is divided in fifteen weeks of the next semester.

Table 1.2: Time table for the second semester.

Weeks Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Modifying the project idea.	■	■	■												
Project requirement and collecting data.			■	■											
Controller design.				■	■	■	■	■							
Parts manufacturing.				■	■	■									
Robot assembly.							■	■							
Testing and experimental results.									■	■	■	■	■	■	■
Results discussion and conclusion.														■	■

1.8 Cost table

Table 1.3 shows the cost of the project, where the project is funded by the Deanship of Graduate Studies and Scientific Research in Palestine Polytechnic University.

Table 1.3: Cost table of the project.

Tools and device	number	Piece price	Price
Servo motor with driver.	1	250	250
Stepper motor.	1	16	16
Workspace.	1	70	70
Gripper.	1	20	20
Rack and pinion.	1	30	30
Time belt and pulleys.	2	25	50
Cables, Wires	1	10	10
Finishing	1	20	20
Bearing.	4	5	20
Servo motor “MG995R”	3	15	45
Raw materials.	1	20	20
Parts manufacturing.			230
	JOD		781
	NIS		3905

2

Chapter 2

Mathematical model

2.1 Direct kinematics

The goal of deriving the direct kinematics is to determine the pose (position and orientation) of the end-effector as a function of the joint variables, with respect to the base frame.

An x-y-z frame will be attached to each joint to describe its pose with respect to the base frame. There is no unique way to attach these frames. A general, systematic method called Denavit-Hartingberg convention will be used to attach these frames [8].

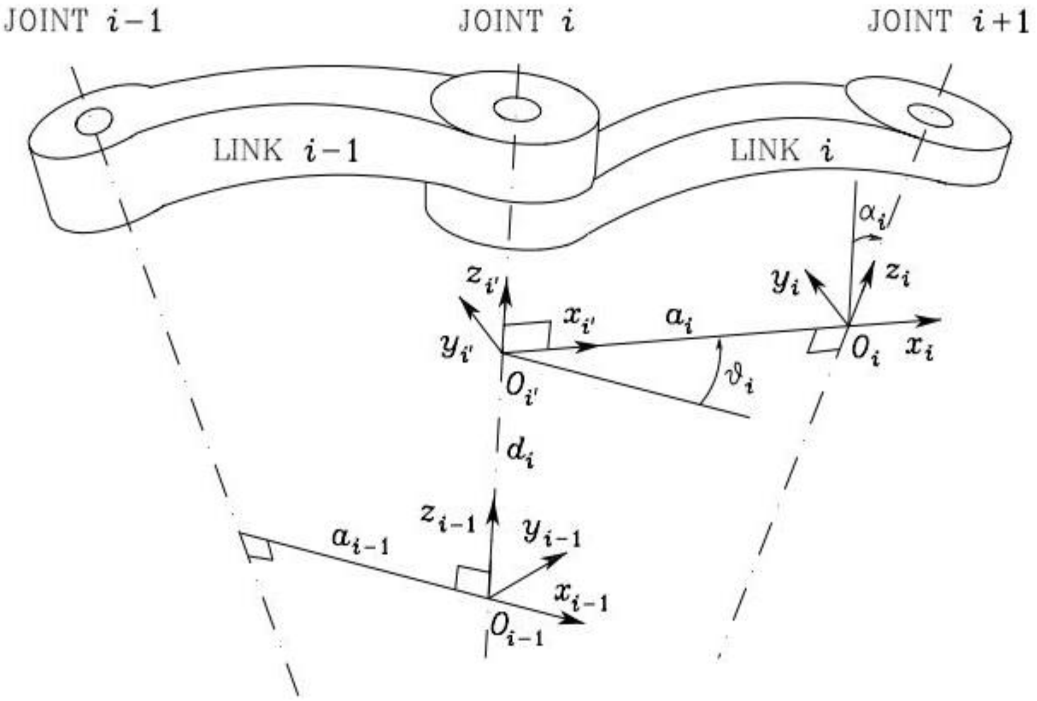


Figure 2.1: Denavit-Hartenberg kinematics parameters. [8]

Referring to Figure 2.1, axis i denotes the joint axis that is connecting the Link $i-1$ with Link i.

D-H convention adopted to define Link frame i based on these rules [8]:

- Choose axis zi along the axis of Joint $i+1$.
- Locate the origin $O_{i'}$ at the intersection of axis z_i with the common normal to axes z_{i-1} and z_i.

  Also, locate $O_{i'}$ at the intersection of the common normal with axis z_{i-1}.
- Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from Joint i to Joint $i-1$.
- Choose axis y_i so as to complete a right-handed frame.

After that, the position and orientation of Frame i with respect to Frame $i-1$ are specified by the following parameters:

a_i distance between O_i and $O_{i'}$

d_i coordinate of $O_{i'}$ along z_{i-1},

α_i angle between axes z_{i-1} and z_i about axis x_i to be taken positive when rotation is made counter-clockwise, and ϑ_i angle between axes x_{i-1} and x_i about axis z_{i-1} to be taken positive when rotation is made counter-clockwise.

The total transformation for each frame to its previous frame is given by:

$$A_i^{i-1}(q_i) = \begin{bmatrix} c_{\vartheta i} & -s_{\vartheta i}c_{\alpha i} & s_{\vartheta i}s_{\alpha i} & a_i c_{\vartheta i} \\ s_{\vartheta i} & c_{\vartheta i}c_{\alpha i} & -s_{\vartheta i}c_{\alpha i} & a_i s_{\vartheta i} \\ 0 & S_{\alpha i} & c_{\alpha i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where q_i is Joint i variable, $c_{\vartheta i}$=cos ϑ_i, $s_{\vartheta i}$=sin ϑ_i, $c_{\alpha i}$=cos α_i and $s_{\alpha i}$=sin α_i.

For a 4-DOF SCARA robot the attached frames are shown in Figure 2.2.

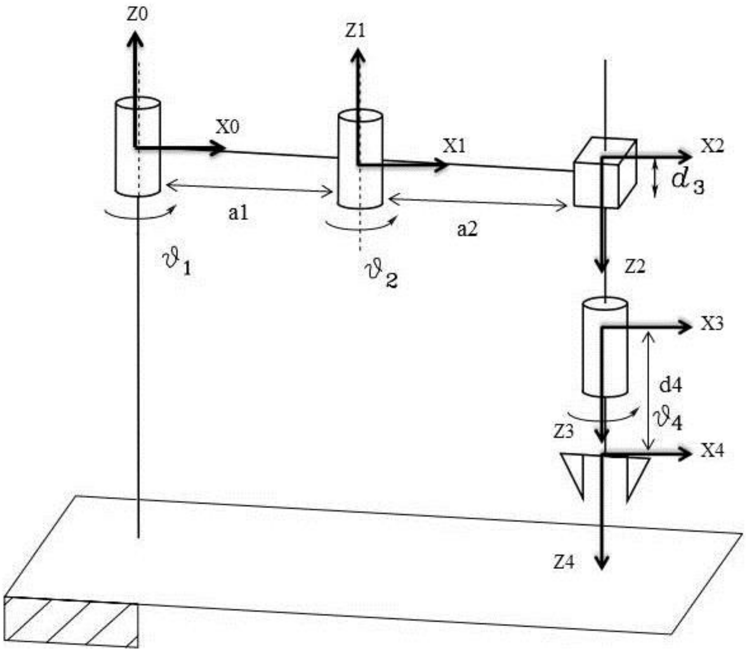


Figure 2.2: SCARA robot joint frames.

The DH parameters are specified in Table 2.1.

Table 2.1: DH parameters for the SCARA robot.

Link	ϑ_{i}	d_i	α_i	a_i
1	ϑ_1	0	0	a_1
2	ϑ_2	0	π	a_2
3	0	d_3	0	0
4	ϑ_4	d_4	0	0

The homogeneous transformation for that end-effector to the base frame given by:

$$T_4^0 = A_1^0 A_2^1 A_3^2 A_4^3; \tag{2}$$

$$A_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

$$A_2^0 = \begin{bmatrix} c_{12} & s_{12} & 0 & a_2c_{12} + a_1c_1 \\ s_{12} & -c_{12} & 0 & a_2s_{12} + a_1s_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

$$A_3^0 = \begin{bmatrix} c_{12} & s_{12} & 0 & a_2c_{12} + a_1c_1 \\ s_{12} & -c_{12} & 0 & a_2s_{12} + a_1s_1 \\ 0 & 0 & -1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

$$T_4^0 = \begin{bmatrix} c_{12-4} & s_{12-4} & 0 & a_2c_{12} + a_1c_1 \\ s_{12-4} & -c_{12-4} & 0 & a_2s_{12} + a_1s_1 \\ 0 & 0 & -1 & -d_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

where c_1=$\cos\vartheta_1$, s_1=$\sin\vartheta_1$, c_2=$\cos\vartheta_2$, s_2=$\sin\vartheta_2$, c_{12}=$\cos(\vartheta_1 + \vartheta_2)$, s_{12}=$\sin(\vartheta_1 + \vartheta_2)$, c_{12-4}=$\cos(\vartheta_1 + \vartheta_2 - \vartheta_4)$ and s_{12-4}=$\sin(\vartheta_1 + \vartheta_2 - \vartheta_4)$.

2.2 Inverse kinematics

The inverse kinematics problem aims to determine the joint variables due to a given end-effector position and orientation. The importance of solving such a problem is to determine the joint variables to achieve desired missions based on specified end-effector poses.

A geometrical approach can solve the inverse kinematics problem. This is suitable for a simple structured robot, but it is preferable to use an analytical approach for complex structured robots.

The joint variables in this SCARA robot are ϑ_1, ϑ_2, d_3 and ϑ_4. The desired end-effector position and orientation in the cartesian space is given by vector P and rotation matrix R_4^0, respectively.

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, \tag{7}$$

$$R_4^0 = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{8}$$

The specified desired homogenous transformation is given by:

$$T_4^0 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The inverse problem is given by:

$$\begin{bmatrix} c_{12-4} & s_{12-4} & 0 & a_2c_{12} + a_1c_1 \\ s_{12-4} & -c_{12-4} & 0 & a_2s_{12} + a_1s_1 \\ 0 & 0 & -1 & -d_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The solution of the Joint 3 variable d_3 is given by:

$$d_3 = d_4 - P_z \tag{11}$$

The solutions of the Joint 2 variable ϑ_2 is given by:

$$c_2 = \frac{P_x^2 + P_y^2 - (a_2^2 + a_1^2)}{2a_2a_1}, \tag{12}$$

$$s_2 = \sqrt{1 - c_2^2}, \tag{13}$$

$$\vartheta_2 = Atan2(s_2, c_2) \tag{14}$$

The solutions of the Joint 1 variable ϑ_1 is given by:

$$s_1 = \frac{(a_1 + a_2c_2)P_y - a_2s_2P_x}{P_x^2 + P_y^2}; \tag{15}$$

where:

$$\vartheta_1 = \sin^{-1}(s_1) \tag{16}$$

The solutions of the Joint 4 variable ϑ_4 is given by:

$$\beta = \vartheta_1 + \vartheta_2 - \vartheta_4, \tag{17}$$

$$\vartheta_4 = \vartheta_1 + \vartheta_1 - \beta\ ; \tag{18}$$

where β is the orientation angle of the end-effector around z_4 with respect to the base frame.

The validations of inverse kinematic is implemented using MATLAB through Peter Cork's Robotics toolbox. Arbitrary end-effector *x*,*y* positions are chosen then angles ϑ_1 and ϑ_2 are computed. After that these angles are applied to the manipulator, the yielded end-effector *x*,*y* positions must be equal to the desired end-effector *x*,*y* positions [9], see Figure 2.3 and Figure 2.4 ,for these numerical example.

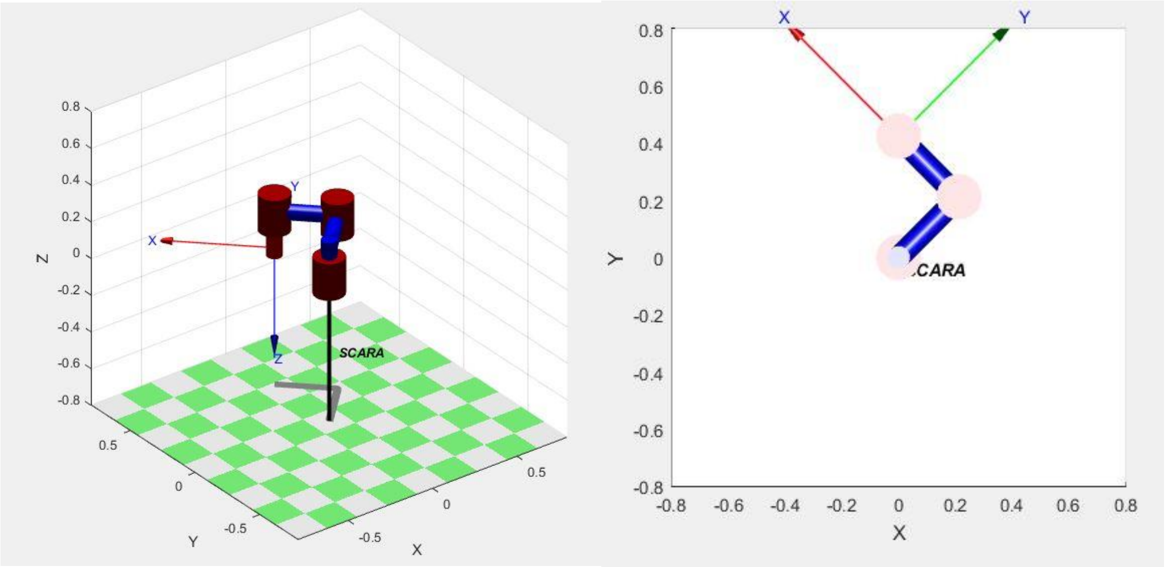


Figure 2.3: (0,424) mm position with $\boldsymbol{\vartheta_1 = 45°, \vartheta_2 = 90°}$

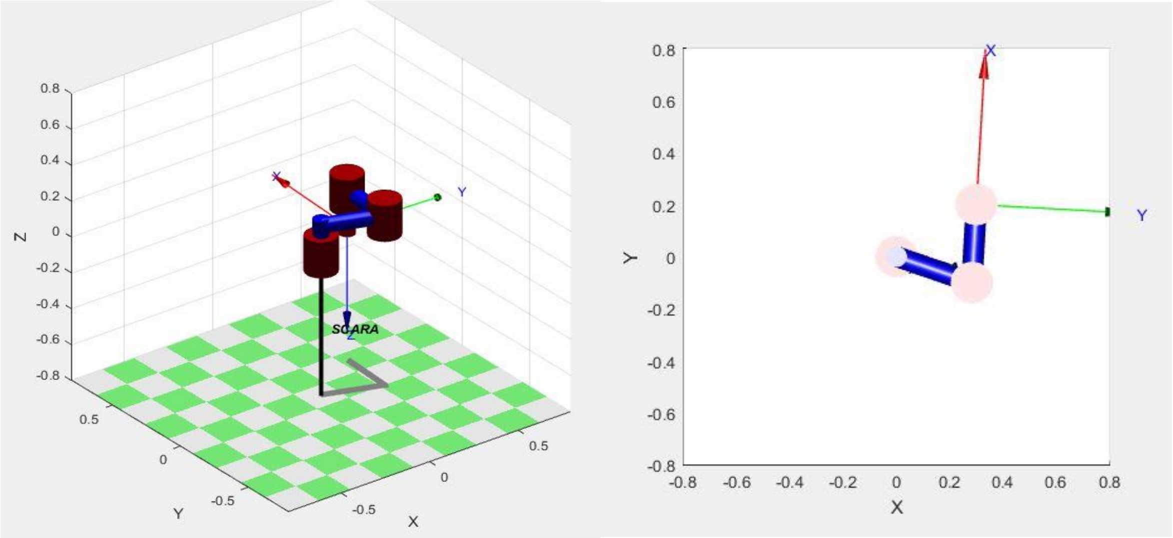


Figure 2.4: (300,200) mm position with $\boldsymbol{\vartheta_1 = -19.37°, \vartheta_2 = 106.12°}$

2.3 Differential kinematics

Differential kinematics give the relationship between the joint velocities and the corresponding end-effector linear and angular velocities [8].

$$\begin{bmatrix} \dot{\boldsymbol{P}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}\,\dot{q}\,, \tag{19}$$

$$\dot{q} = [\dot{\vartheta}_1\;\dot{\vartheta}_2\;\dot{d}_3\;\dot{\vartheta}_4]^T; \tag{20}$$

where $\dot{\boldsymbol{P}}_e$ and $\boldsymbol{\omega}_e$ are the end-effector linear and angular velocities respectively, $\mathbf{J}$ is the Jacobian matrix and $\dot{\vec{q}}$ is the joint velocities vector.

The Jacobian matrix for the SCARA robot is given by:

$$\mathbf{J}(q) = \begin{bmatrix} z_0 \times (p_4 - p_0) & z_1 \times (p_4 - p_1) & z_2 & z_3 \times (p_4 - p_3) \\ z_0 & z_1 & \mathbf{0} & z_3 \end{bmatrix} \tag{21}$$

From the homogenous transformation computation:

$$z_0 = z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, z_2 = z_3 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, p_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, p_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix}, p_2 = \begin{bmatrix} a_2 c_{12} + a_1 c_1 \\ a_2 s_{12} + a_1 s_1 \\ 0 \end{bmatrix} \tag{22}$$

$$p_3 = \begin{bmatrix} a_2 c_{12} + a_1 c_1 \\ a_2 s_{12} + a_1 s_1 \\ -d_3 \end{bmatrix}, p_4 = \begin{bmatrix} a_2 c_{12} + a_1 c_1 \\ a_2 s_{12} + a_1 s_1 \\ d_4 - d_3 \end{bmatrix} \tag{23}$$

Hence,

$$\mathbf{J}(q) = \begin{bmatrix} -(a_2 s_{12} + a_1 s_1) & a_2 s_{12} & 0 & 0 \\ a_2 c_{12} + a_1 c_1 & -a_2 c_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \tag{24}$$

2.4 Dynamic model

The controller for the manipulator will be designed based on the dynamic model of that manipulator, which provides a description of the relationship between the joint actuator torques and the motion of the structure [8]. There are some methods to derive the equation of motion for such a robot, like Newton Euler's formulation, Lagrange formulation …etc. The general equation of motion is shown in Eq(25); where $M(q)$ is the mass matrix, $C(q, \dot{q})$ is the Coriolos and centrifugal matrix, $g(q)$ is the gravity vector, F_v friction coefficients matrix and $\tau(q)$ torque vector [8].

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v\dot{q} + g(q) = \tau(q) \tag{25}$$

Noting that the mass matrix is configuration dependent (function of joints position q). This mathematical dependency emphasizes the physical fact that the robot arm has different inertia based on its configuration [10].

Lagrange formulation is used to derive the SCARA robot dynamic in this project, for its simple systematic derivation for such a complex system.

Lagrange equation of motion is given by [8] :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \,, \quad i = 1,2,3,4, \tag{26}$$

where L is the Lagrangian, which is given by:

$$L = K - P \tag{27}$$

Where K and P are the total kinetic energy and the total potential energy of the manipulator, respectively. To compute the total kinetic energy of the manipulator, it is needed to compute the kinetic energy for each link and each motor individually [8].

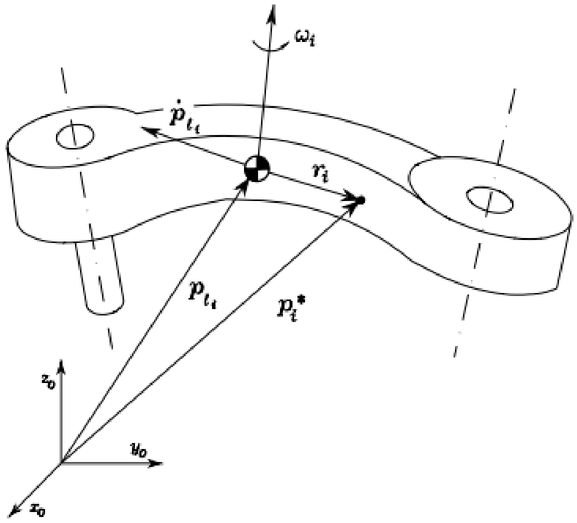


Figure 2.5: Kinematic description of Link *i*. [8]

The velocity of Link i is considered by:

$$\dot{\vec{P}}_i^* = \dot{\vec{P}}_{li} + \omega_i \times r_i, \tag{28}$$

where $\dot{\vec{P}}_{li}$ is the linear velocity for the center of mass of the link with respect to the base frame and $\omega_i \times r_i$ represents the relative velocity of each point on the line to the center of gravity of that link with respect to frame i, ω_i represents the angular velocity of the link [8].

Hence the total kinetic energy for each link will be translational compounded with rotational kinetic energy.

Link i kinetic energy is given by:

$$K_i = \frac{1}{2} m_{Li} \dot{P}_{li}^T \dot{P}_{li} + \frac{1}{2} \omega_i^T J_{Li} \omega_i \tag{29}$$

The differential kinematics take a place here, to map between the joint/s velocity that cause the link move and the translational velocity of the center of gravity of that link also the angular velocity of that link.

Link i potential energy is given by:

$$U_{li} = -m_{Li} g^T P_{li} \,, \tag{30}$$

$$g = [0 \quad 0 \quad -9.8]^T \tag{31}$$

After applying the Lagrange formulation and rearranging the terms in the form that is given by Eq(25). The yielded model is expressed and detailed below.

The mass matrix is given by:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & 0 & M_{14} \\ M_{21} & M_{22} & 0 & M_{24} \\ 0 & 0 & M_{33} & 0 \\ M_{41} & M_{42} & 0 & M_{44} \end{bmatrix}; \qquad (32)$$

where:

$$M_{11} = m_{L1}l_1^2 + \ m_{L2}(l_2^2 + a_1^2 + 2l_2a_1c_2) + \ m_{L3+L4}(a_2^2 + a_1^2 + 2a_1a_2c_2) + \ J_{L1} + \ J_{L2} + J_{L4}\,,$$

$$M_{12} = M_{21} = \ m_{L2}(l_2^2 + l_2a_1c_2) + \ m_{L3+L4}(a_2^2 + a_2a_1c_2) + \ J_{L2} + \ J_{L4}\,,$$

$$M_{22} = \ m_{L2}l_2^2 + m_{L3+L4}a_2^2 + J_{L2} + J_{L4}\,,$$

$$M_{14} = M_{41} = \ -J_{L4}\,,$$

$$M_{24} = M_{42} = \ -J_{L4}\,,$$

$$M_{33} = m_{L3+L4}\,,$$

$$M_{44} = \ J_{L4}$$

The Coriolis and centrifugal matrix is given by:

$$C(q,\dot{q}) = \begin{bmatrix} A\dot{\vartheta}_2 & A(\dot{\vartheta}_1 + \dot{\vartheta}_2) & 0 & 0 \\ -A\dot{\vartheta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \qquad (33)$$

where:

$$A = -m_{L2}l_2a_1s_2 + -m_{L3+L4}a_1a_2s_2 \qquad (34)$$

The gravitational vector by:

$$g(q) = \begin{bmatrix} 0 \\ 0 \\ -m_{L3+L4}g_o \\ 0 \end{bmatrix}, \qquad (35)$$

where:

$$g_o = 9.8\frac{m}{s^2} \qquad (36)$$

Here, all viscous and dry frictions are neglected.

3

Chapter 3

Mechanical Design

3.1 Mechanical conceptual design

Several mechanical conceptual designs were suggested for the SCARA robot. The difference between them is the forces distribution, hence they do not have the same factor of safety. They were designed using SOLIDWORKS software [2].

3.1.1 The proposed conceptual design

The exploited views for the robot is detailed in Figure 3.1-3.5. Table 3.1 contain the names of the numbered parts in the mentioned figures.

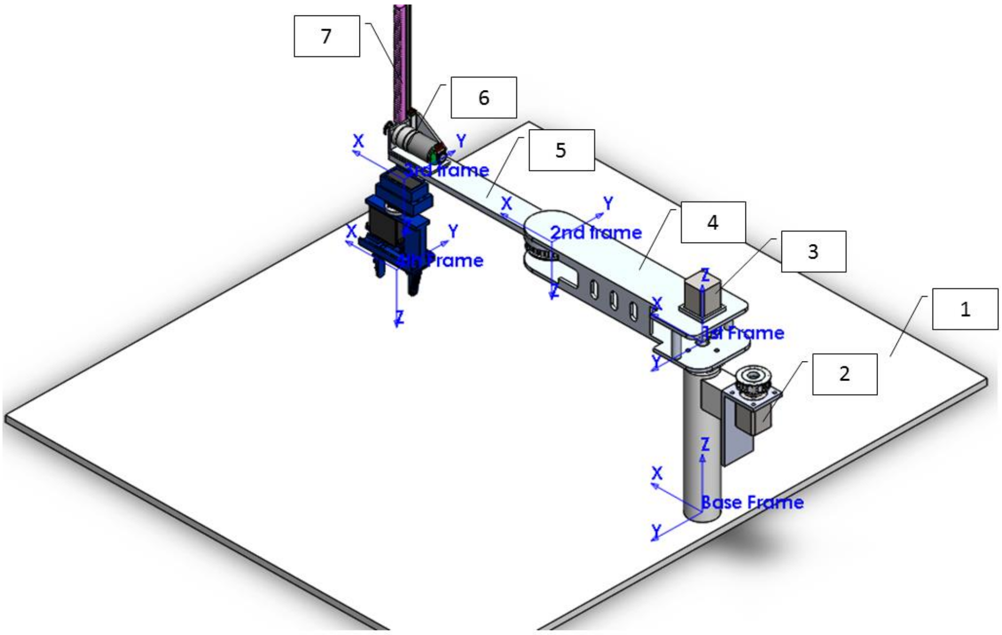


Figure 3.1: Isometric view

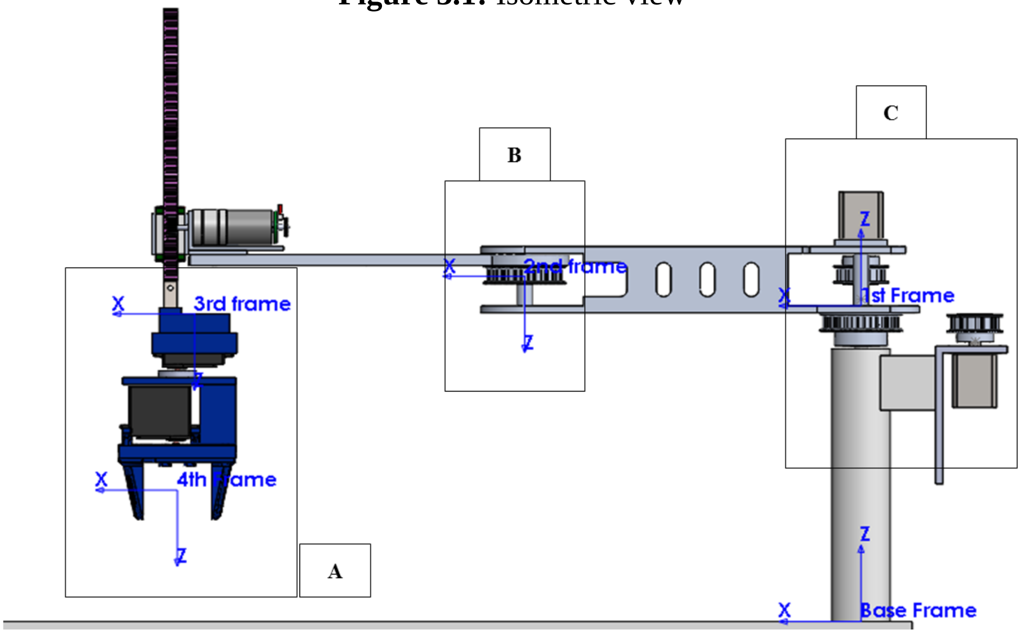


Figure 3.2: Front view.

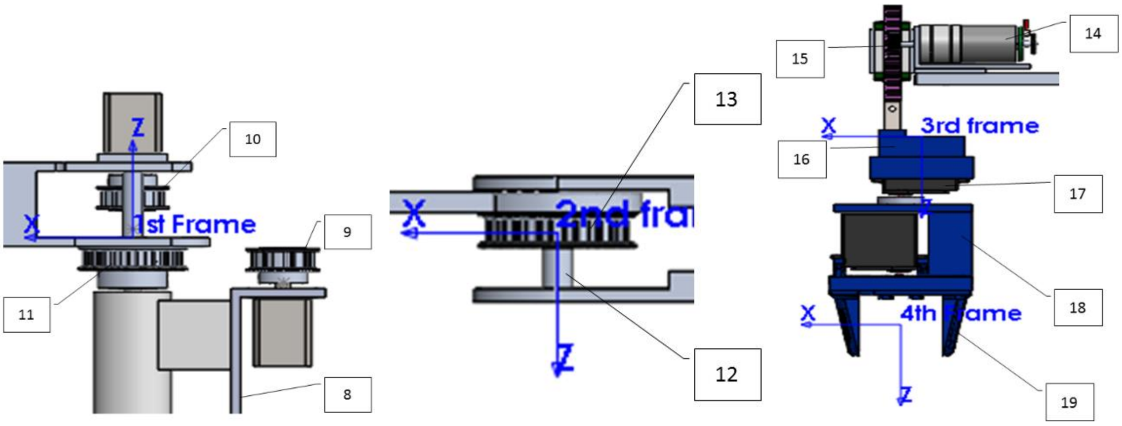


Figure 3.3: Detailed sections A, B and C in the front view.

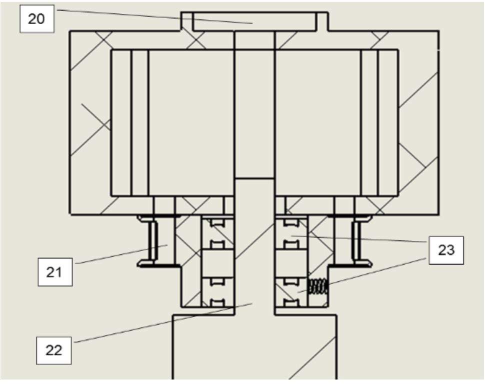


Figure 3.4: Cross section for the section A in the front view.

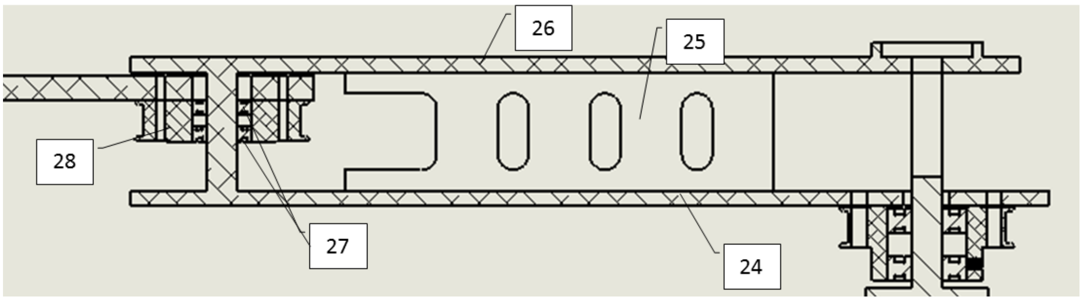


Figure 3.5: Cross section for the section A in the front view.

Table 3.1: The numeric parts

Part Number	Part Name
1	Work-area ($800\ mm * 700mm$).
2	First Link actuator (Delta AC-servomotor)
3	Second Link actuator (Stepper-motor)
4	First Link
5	Second Lenk
6	Third Link actuator (DC-servomotor)
7	Third Link
8	Stationary base for the First Link actuator
9	Driving pulley for the First Link (10 teeth)
10	Driving pulley for the Second Link (20 teeth)
11	Driven pulley for the First Link (40 teeth)
12	Stationary rod for The Second Link
13	Driven pulley for the Second Link (20 teeth)
14	Third Link actuator (DC-servomotor)
15	Rack and pinion for The Third prismatic joint
16	Housing of the servomotor for the Wrist joint
17	End-effector wrist actuator (DC-servomotor)
18	The main support for the End-effector mechanism
19	Moving parts of the end-effectors for grabbing
20	The Housing for the Second Link actuator
21	The Driven pulley of the First Link
22	Rod of the manipulator base
23	Double ball-bearings for the First Link
24	Lower portion of the First Link
25	Double side support for the upper and lower portions
26	Upper portion of the First Link
27	Double ball-bearings for the Second Link
28	Driving pulley for the Second Link (20 teeth)

Where Parts 15 and 16 will be made of 3D printed plastic. Part 9 detailed dimensions will be determined after buying the servomotors, it will be made from sheet metal to be attached to part 2 and formulated to be a fixture for the first servomotor.

The axis of rotation of the Second Link actuator will be collinear with the first axis of rotation as shown in Figure 3.4 and Figure 3.5, the reason of this choice is to reduce the inertia that the first link actuator must rotate where the inertia of a body around an axis that has an offset from its axis of rotation given by the parallel axis theorem shown in Eq(37) [11], here $d = 0$. Moreover, this choice will yield a zero-bending stress on the base of the First Link due to actuator weight.

$$J = \bar{J} + md^2; \qquad (37)$$

Where J is the mass moment of inertia about an axis (here the axis of rotation of the First Link), $\bar{J}$ is the mass moment of inertia about an axis that pass through the center f mass of that body (here the axis of rotation of second link actuator), m is the mass of that body and d is the separation between these axes [11].

Summary of the final design

The power transmitters to rotate the first link and the second link are timing belts, because they do not stretch appreciably due to the steel wire within to take the tension load, have efficiencies in the range of $(97\% - 99\%)$, require no lubrication, and quitter than other power transmitters so that they are preferable for precision-drive requirements [8]. The third DOF (prismatic joint) performed using rack and pinion mechanism.

3.1.2 Another proposed conceptual design

The deference here that the Second Link actuator axis of rotation has an offset from the axis of rotation of the first link, Figure 3.6 shows that configuration. In addition of the advantages of decreasing the inertia and bending stress proposed by the final design, it is also increased the second area moment of inertia of the first link by increasing the height of the side supports (the offset between the upper and lower portions), this will yield a decreasing in the bending stress.

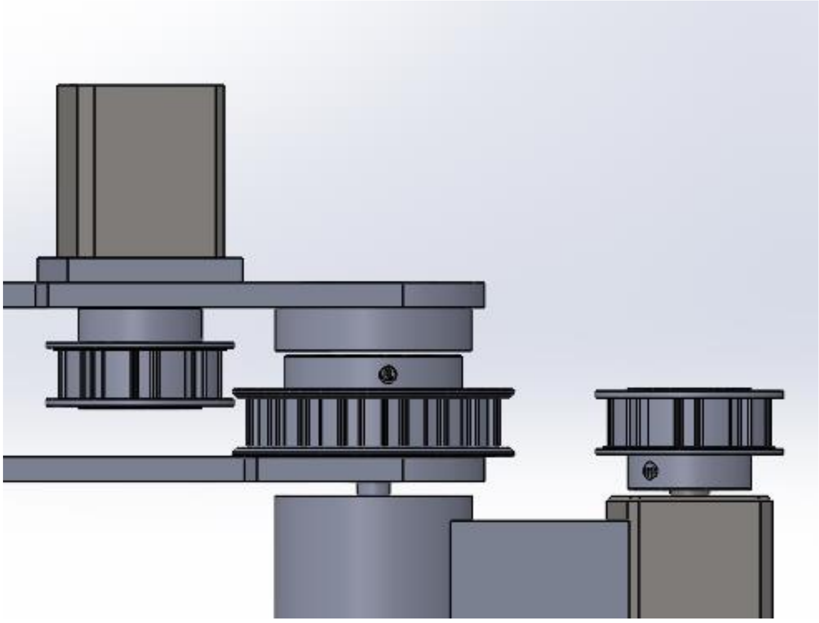


Figure 3.6: The second link actuator configuration for the first conceptual design.

3.2 Mechanical design

The first consideration for designing the robot arm is to realize the system specifications (range, working envelop and load capacity), these specifications will be the starting point, constraints and limitations for the final design [11].

The working envelope is configured carefully based on the possibilities and capabilities to implement this project. On one hand, an assumption was made to design the working envelop is to locate from the base frame of the manipulator such that the farthest object that the manipulator has to grab is equal to $35\ cm$. On the other hand, the manipulator must reach that location with large manipulation capability. The manipulability of the manipulator describes how much the joint velocities could develop an end-effector velocity [8]. A mathematical interpretation is developed to give a measure for the *manipulability* by taking advantage of the deferential kinematics developed before [8]. The *manipulability* of the robot given by:

$$w(q) = \sqrt{|J(q)\,J^T(q)\,|}, \tag{38}$$

For the previously modeled SCARA robot the measure is given by:

$$w(q) = a_1 a_2 \sin\theta_2\,; \tag{39}$$

When $\vartheta_2 = \pm 90^{\circ}$, $w(q)$ will be the maximum measure. When $\vartheta_2 = 0^{\circ}, 180^{\circ}$ (singularity postures) then $w(q) = 0$. Choose first link's length to be $23\ cm$ also second link to be $23\ cm$, i.e. when $\vartheta_2 = \pm 58.5^{\circ}$ the farthest point will be $35\ cm$ away from the base frame (the boundary of the assumed working envelope). The measure at the boundary of the envelop will be

$$w(q) = 0.85\ a_1 a_2 \tag{40}$$

This measure is admissible for such missions. Hence the links length will be $23\ cm$ for both links.

The payload object that the manipulator has to pick and place at a time will weigh $0.49\ N$ ($50\ gm$).

3.2.1 Links detailed design:

Aluminum 2024 alloy is selected to be the metal for the links, because of its light weight and good stiffness, its yield strength is $(76\ MPa)$ and *Steel 1020 alloy* for the main support of the manipulator, its yield strength is $(351\ MPa)$. The manipulator mechanical structure must not exceed a deflection of $0.5\ mm$ in order not to affect the accuracy of positioning the end-effector. The design process will be applied when the mechanical structure is in its critical configurations, in other words the configurations, those yield the maximum stresses on the structure. The forces that considered in design process are both static forces and inertial forces, which arise when the manipulator is accelerating or decelerating. The SCARA robot configurations may affect the base of the First Link as bending stress or compound stress (bending and torsion stresses). The bending stress is given by

$$\sigma_x = \frac{M_y z}{I_y} + \frac{M_z y}{I_z}, \tag{41}$$

$$I_y = \int z^2 dA, \tag{42}$$

$$I_z = \int y^2 dA, \tag{43}$$

where σ_x is the stress along the $x - axis$ (neutral axis), M_y is the bending moment around the $y - axis$ (it acts in $xy - plane$), I_z is the second moment of area around the $z - axis$ and I_y is the second moment of area around the $y - axis$ [12]. The structure of the robot links shows that $I_z \gg I_y$ see Figure 3.7, therefore $\frac{M_z y}{I_z}$ could be negligible. The ratio between $I_z \; and \; I_y$ is given by for second link cross-section:

$$\frac{I_z}{I_y} = \frac{y^2}{z^2} = 25, \tag{44}$$

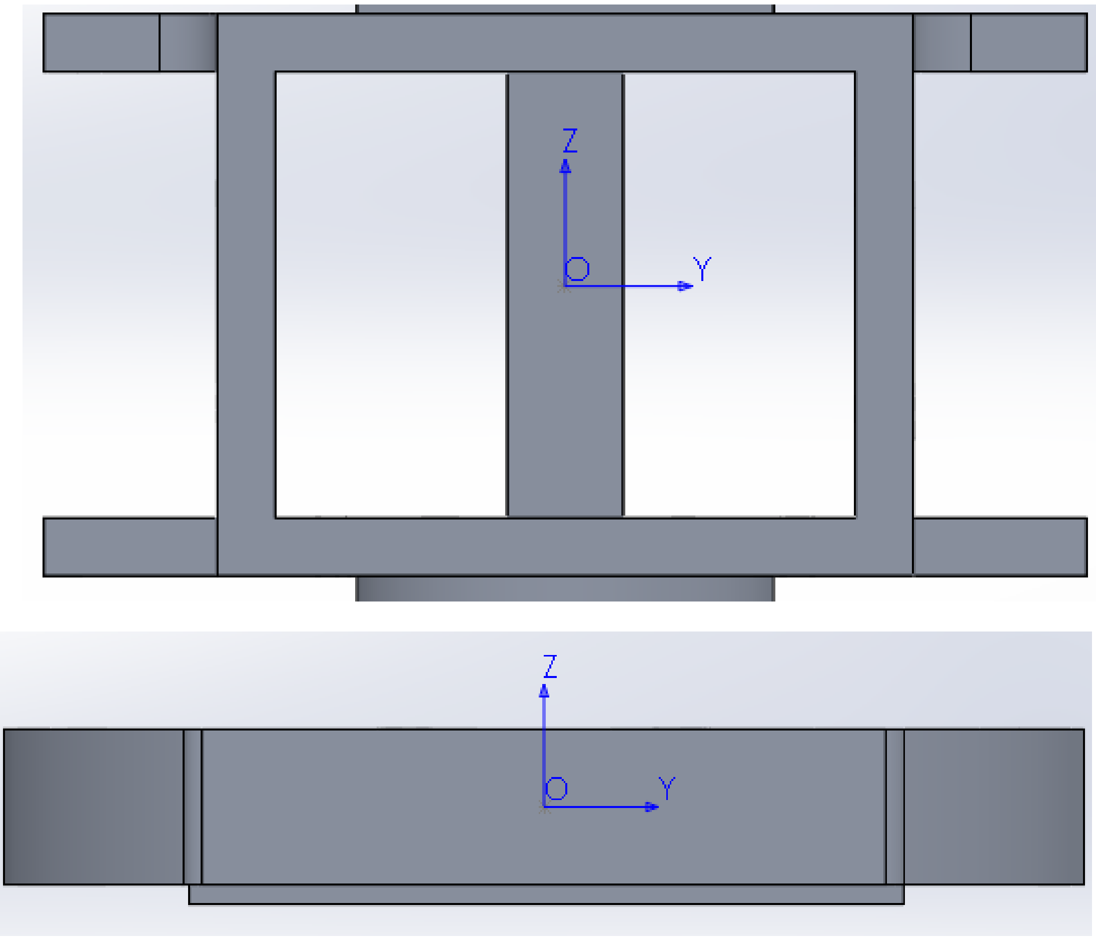


Figure 3.7: First Link & Second Link cross-sections.

The maximum torsional shear stress in rectangular cross-section $(b \times c)$ is given by

$$\tau_{max} = \frac{T}{bc^2}(3 + \frac{1.8}{b/c}), \tag{45}$$

Where T is the moment vector that collinear with the neutral $x - axis$ [12].

The strategy that used here to design the links is to assume the thicknesses of the links that assembled in the mechanical conceptual design. Then stresses of critical configurations computed and simulated numerically using SOLIDWORKS software due to complexity of the mechanical structure that detailed in Section 3.1.1.

The thicknesses of the first link members are assumed to be 5 mm, and the second link is assumed to be 8 mm. That's yield a First Link mass of 0.7 Kg and Second Link mass of 0.233 Kg, the end-effector and the payload have a mass of $0.2\ Kg$, the rack of the third joint and its linear guide have a mass of $0.1\ Kg$ and the actuator of the third link have a mass of $0.2\ Kg$.

Hence, the forces that acts on the farthest point on the arm (see Figure 3.8) is given by:

$$W_{static} = (0.2 + 0.2 + 0.1) * 9.8 = 4.9\ N \downarrow,$$

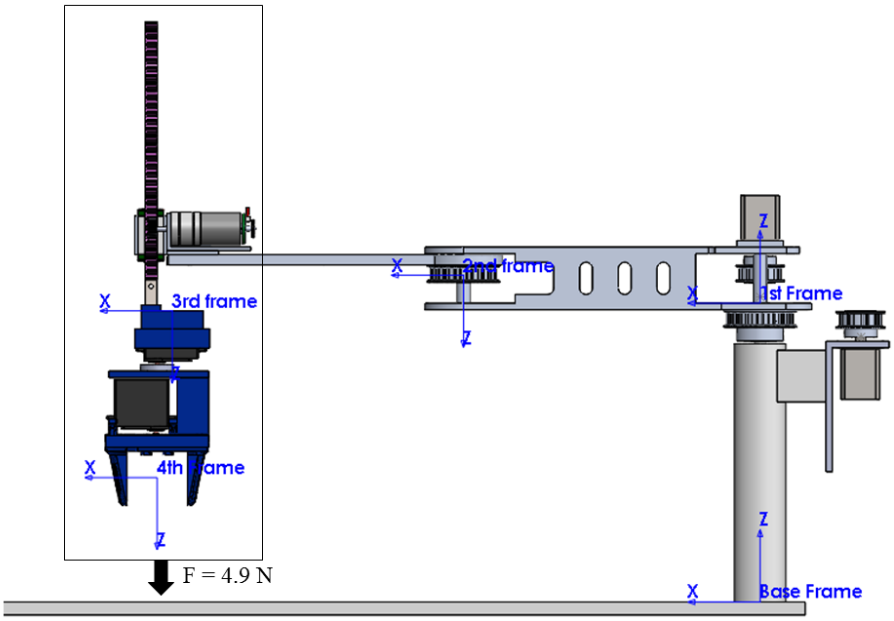


Figure 3.8: Front view with the end-effector mechanisms static load.

The inertial force yielded when the rack (including the end-effector and the payload) accelerating by $10\ cm/s^2$(the worst case) upward given by:

$$W_{inertial} = 0.3 * 0.1 = 0.03\ N \downarrow,$$

$$W_{total} = W_{inertial} + W_{static} = 4.93\ N \approx 5\ N \downarrow;$$

i.e. the stress analysis will be perform using SOLIDWORKS by applying $5\ N \downarrow$ at the end of the arm.

The first configuration and its stress results shown in Figure 3.9 and Figure 3.10:

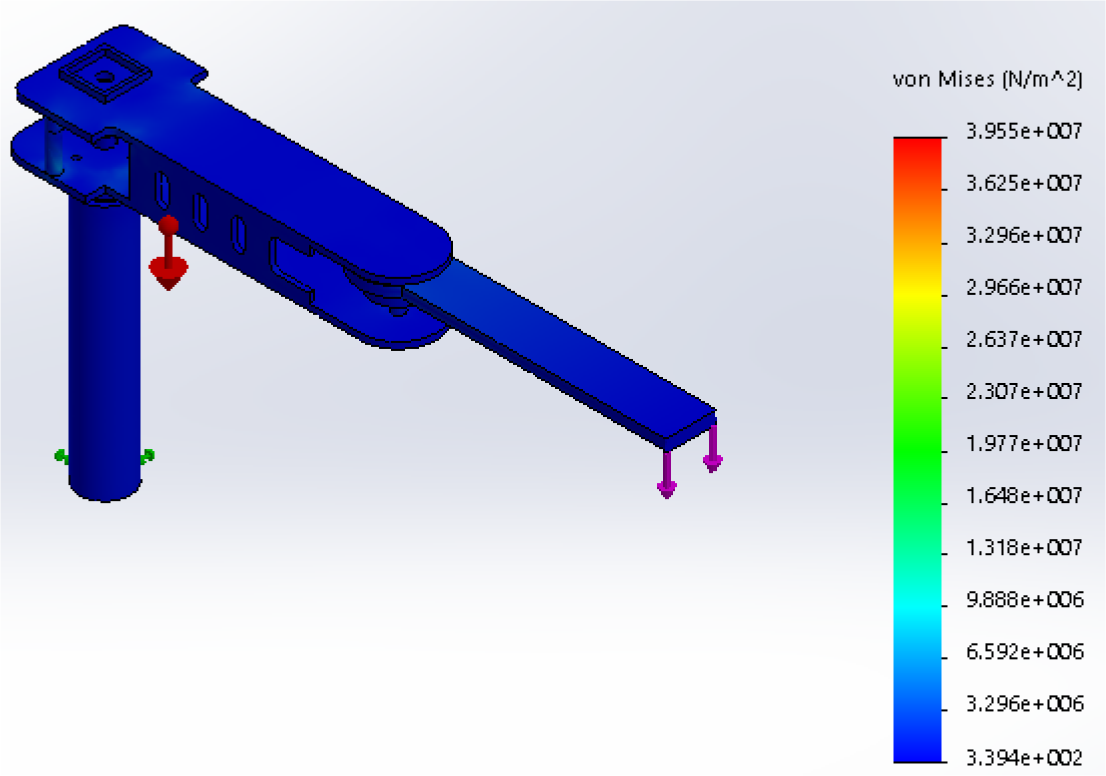


Figure 3.9: Stress result due to the mass of the arm and **5 *N*** ↓ at the end of the arm.

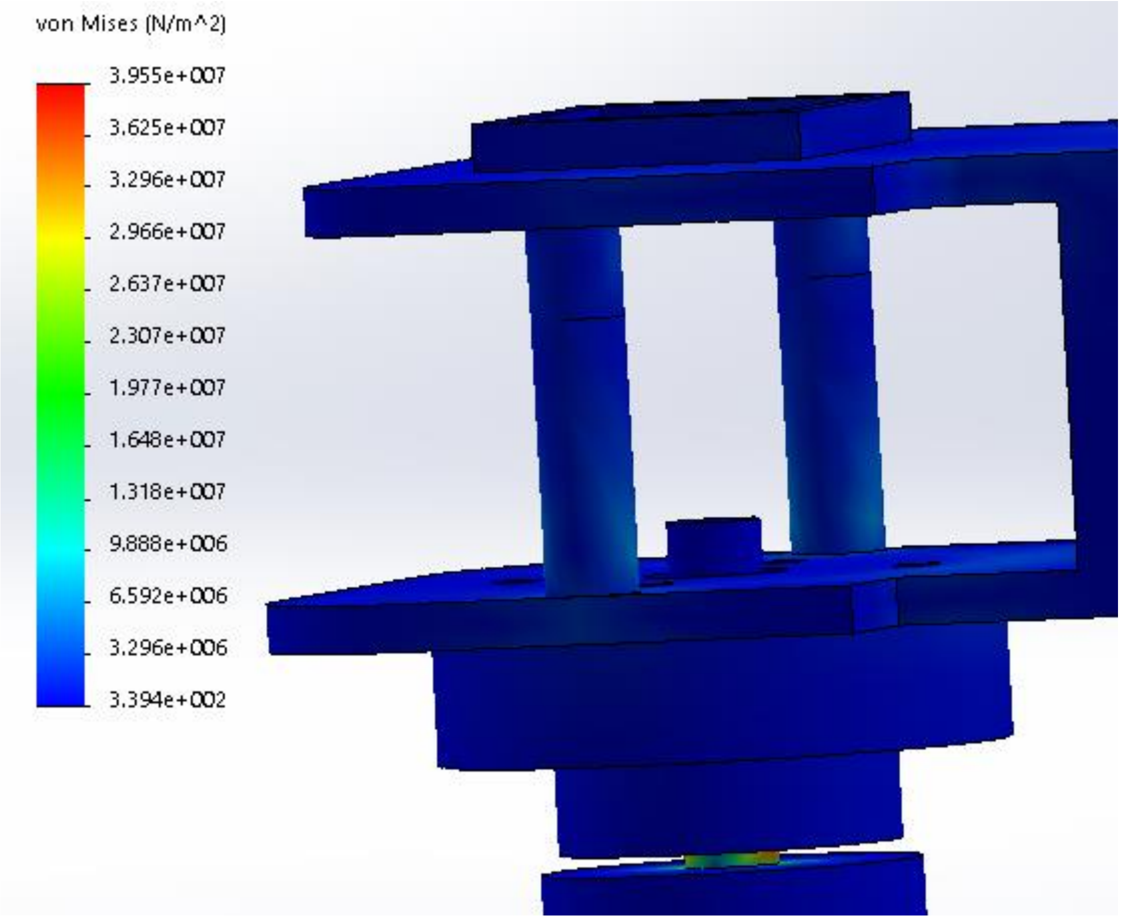


Figure 3.10: Location where the maximum stress occurs.

The results show that the maximum Von Mises stress due to bending $\approx 6.5\ MPa$ in the aluminum links, where in the steel rod$\approx 38\ MPa$.

$$n_{Al} = \frac{75}{6.5} = 11.5,$$

$$n_{steel} = \frac{351}{38} = 9.23,$$

$$n = 9.23;$$

The deflection due to bending shown in the Figure 3.11, and the maximum deflection is $0.46\ mm$. This deflection is acceptable and it does not contract with assumption criteria of the design.

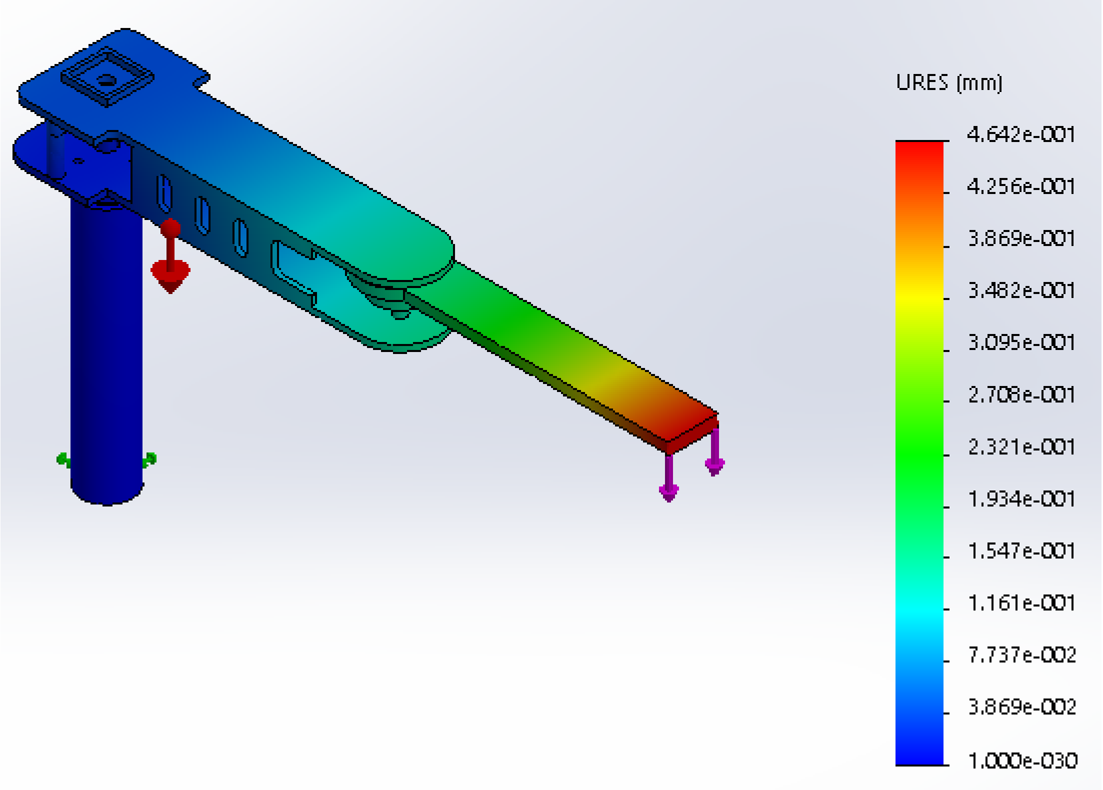


Figure 3.11: The maximum deflection in the First Configuration.

The second configuration assumed to give the maximum possible torsional stress; in this case a compound stress will be applied in the critical point, the stress results are shown in Figure 3.12 and Figure 3.13.

The same inertial and static forces applied to the CAD model, the only matter here is the configuration of the arm.

$$W_{inertial} = 0.3 * 0.1 = 0.03\ N \downarrow,$$

$$W_{static} = (0.2 + 0.2 + 0.1) * 9.8 = 4.9\ N \downarrow,$$

$$W_{total} = W_{inertial} + W_{static} = 4.93\ N \approx 5\ N \downarrow;$$

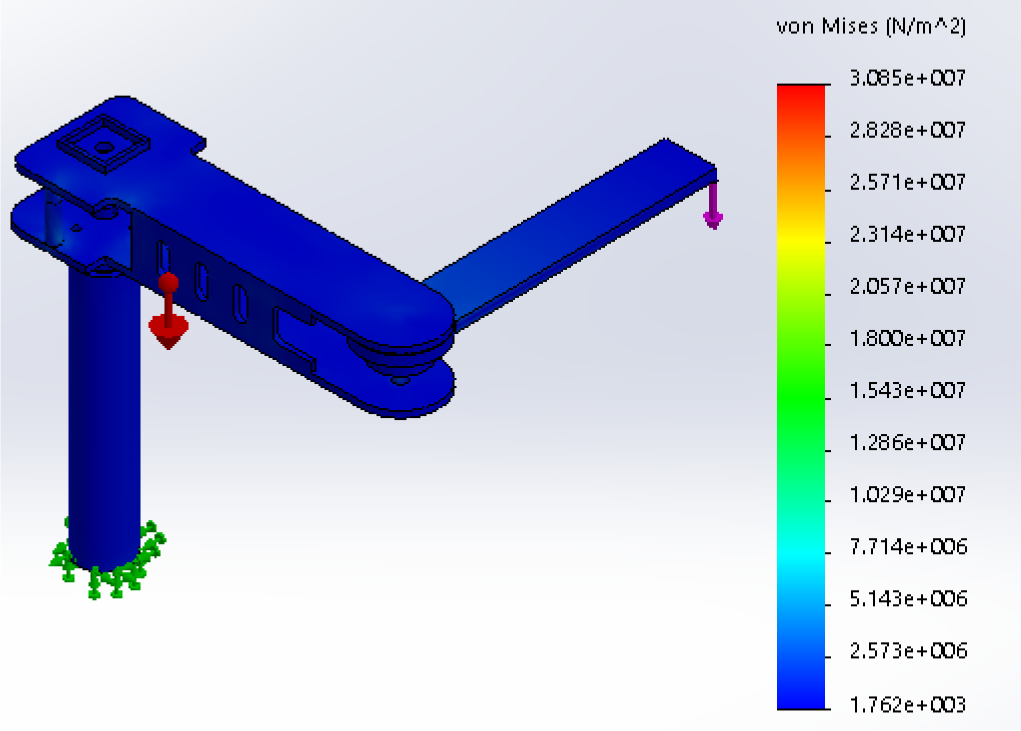


Figure 3.12: Stress result due to the mass of the arm and **5 *N*** ↓ at the end of the arm.

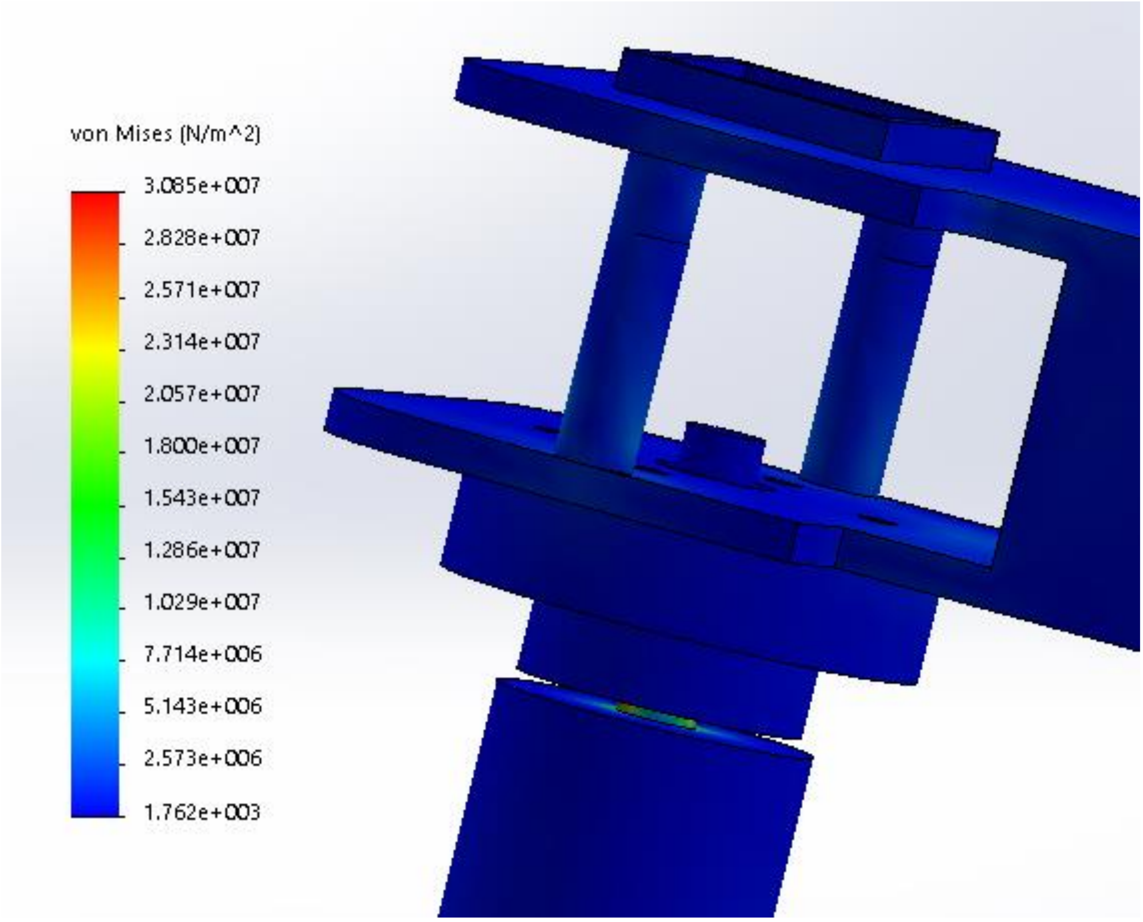


Figure 3.13: Location where the maximum stress occurs.

The results show that the maximum Von Mises stress due to compound stress $\approx 5.3\ MPa$ in the aluminum links, where in the steel rod$\approx 28\ MPa$.

$$n_{Al} = \frac{75}{5.3} = 14.15,$$

$$n_{steel} = \frac{351}{28} = 12.5,$$

$$n = 12.5;$$

The deflection due to compound stress is shown in the Figure 3.15, and the maximum deflection is 0.349 mm. This deflection is acceptable and it does not contract with assumption criteria of the design.

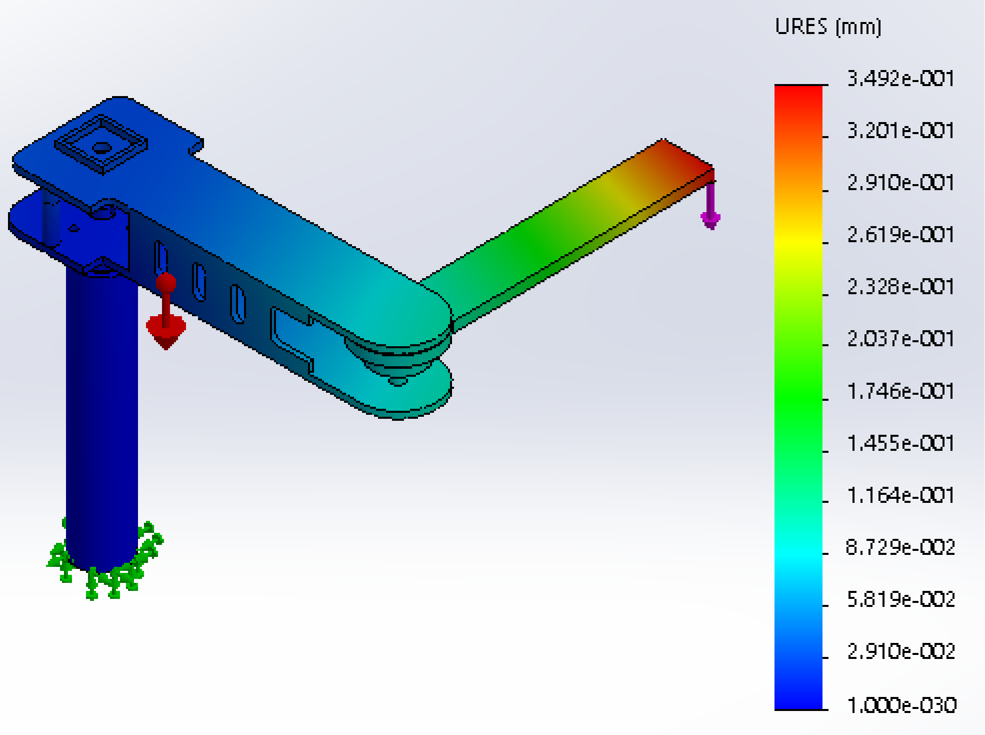


Figure 3.14: The maximum deflection in the Second Configuration.

The assumption that mentioned before that the thickness of the members for the first link is 5 mm and 8 mm for the Second Link is acceptable.

3.2.2 Links and parts dimensions:

The following figures (Figure 3.15 to Figure 3.17) give a datailed drawing for each link and part of the manipulator, where all dimensions in melimeters (mm) and degrees.

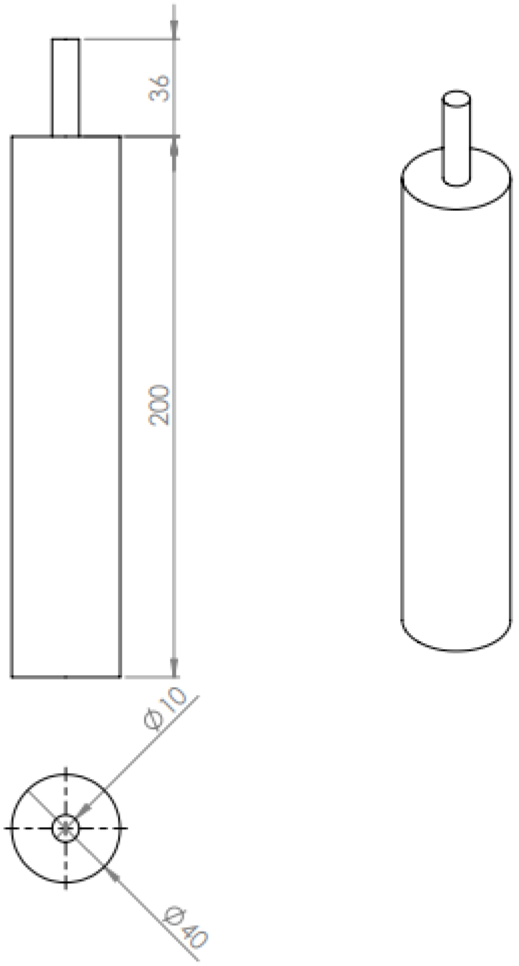


Figure 3.15: The base of the manipulator (the main support), all dimensions in mm.

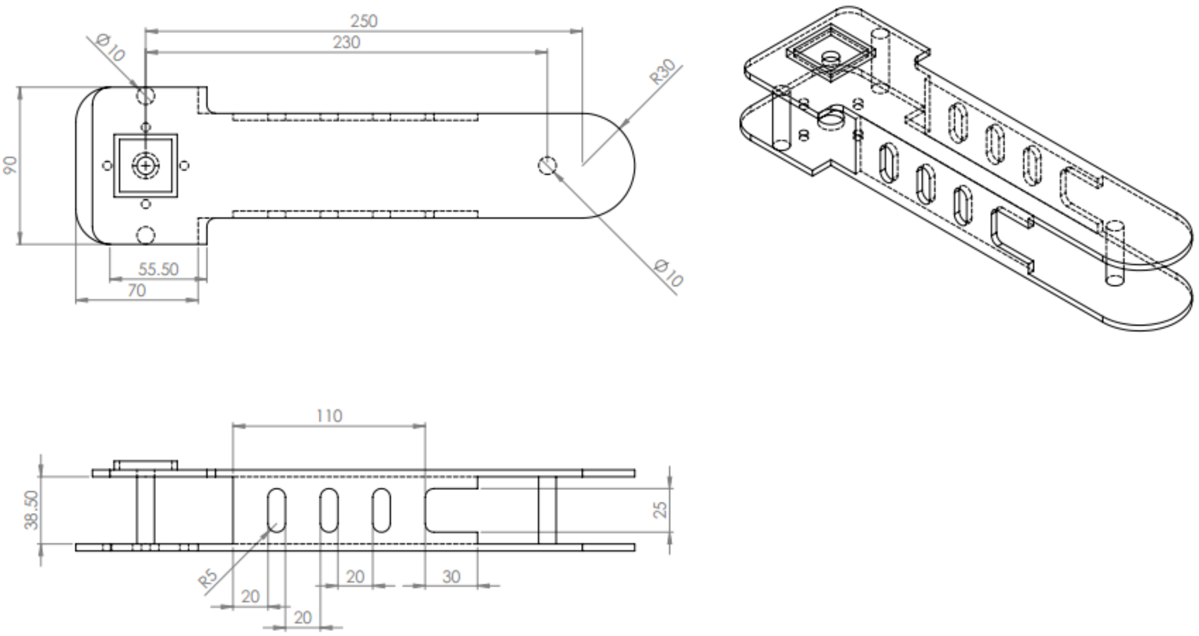


Figure 3.16: The first link, all dimensions in mm.

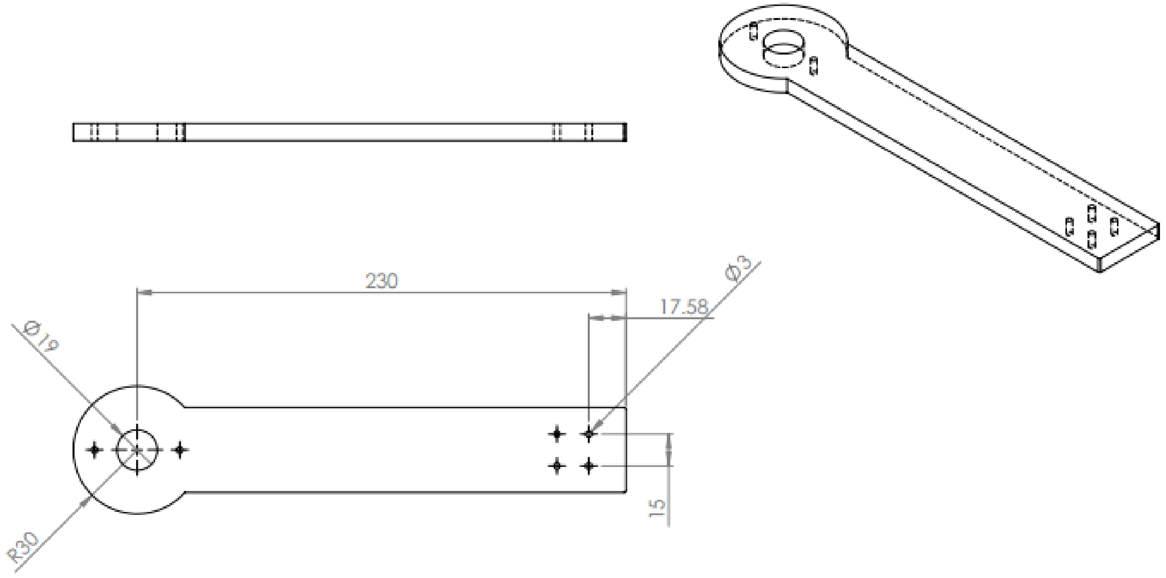


Figure 3.17: The second link, all dimensions in mm.

3.2.3 Mechanical design summary:

Table 3.2 and Table 3.3 give details for the designed parts and the connections between links.

Table 3.2: Parts specifications.

	Mass $[\boldsymbol{Kg}]$	Length $[\boldsymbol{m}]$	Center of Gravity $[\boldsymbol{m}]$	$\boldsymbol{I_{Z-COG}[Kg.m^2]}$
The base	1.806	0.211	(0,0,0.0911)	0.0004
First Link	0.7418	0.25	(0.1108,0,0)	0.0056
Second Link	0.2554	0.25	(0.1078,0,0)	0.0017
Third and Fourth mechanism	0.5	-	-	-

Note that center of gravity (COG) of a link taken with respect to that link frame the developed previously in the forward kinematics. For the base COG taken with respect to the ground frame.

Table 3.3: Bearings between parts.

	Bearing model number	Bore, OD, Thickness(mm)
The base with First Link	SKF – 6200 – 8,SI,NC,8_68	10,30,8
First Link with Second Link	SKF – 61800 – 14,NC,14_68	10,19,5

3.3 Motor sizing

Manipulator parts-specifications have been already detailed in the Section 3.2. Now the motor sizing process is ready to be accomplished.

The general equation of motion that is shown in Eq(25) describes each joint generalized force with respect to robot configuration. Torque for Joint 1, Torque for Joint 2, Joint 3 force and Joint 4 torque shown in Eq(46), Eq(47), Eq(48) and Eq(48), respectively.

$$\tau_1 = M_{11}\ddot{\vartheta}_1 + M_{12}\ddot{\vartheta}_2 + M_{14}\ddot{\vartheta}_4 + A\dot{\vartheta}_2\dot{\vartheta}_1 + A\left(\dot{\vartheta}_1 + \dot{\vartheta}_2\right)\dot{\vartheta}_2 \tag{46}$$

$$\tau_2 = M_{21}\ddot{\vartheta}_1 + M_{22}\ddot{\vartheta}_2 + M_{24}\ddot{\vartheta}_4 - A\dot{\vartheta}_1^2 \tag{47}$$

$$F_3 = M_{33}\ddot{d}_3 - M_{33}g_0 \tag{48}$$

$$\tau_4 = M_{41}\ddot{\vartheta}_1 + M_{42}\ddot{\vartheta}_2 + M_{44}\ddot{\vartheta}_4 \tag{49}$$

where M_{11}, M_{12}, M_{14}, M_{21}, M_{22}, M_{24}, M_{33}, M_{41}, M_{42} and M_{44} are detailed in mass matrix in Section 2.4. Also A is detailed in the Coriolis and Centrifugal matrix in Section 2.4.

The relationship between the joint torques/forces and joint positions, velocities, accelerations is completely clear by the dynamic model. The strategy used to size the motors is to design critical trajectories with deferent scenarios, then applying the yielded trajectories on Eq(46), Eq(47), Eq(48) and Eq(49) to determine the needed torque for each joint to actuate the manipulator along that trajectory.

First scenario:

The end-effector must move from point $(0.46,0,0)m$ to point $(-0.305, 0.305,0)m$ in 1.5 seconds as shown in Figure 3.18. Where $\vartheta_1$and ϑ_2 are $(0° \ to \ 120°)$ and $(0° \ to \ 30°)$, respectively, they are computed through inverse kinematics.

A trapezoidal velocity profile for the trajectory is assigned, which impose a constant acceleration and deceleration in start phase and arrival phase, respectively [8] (see Figure 3.19 and Figure 3.20). Hence the motion planner will be able to assign the accelerations and velocities that each motor has to deliver to track the assigned trajectory.

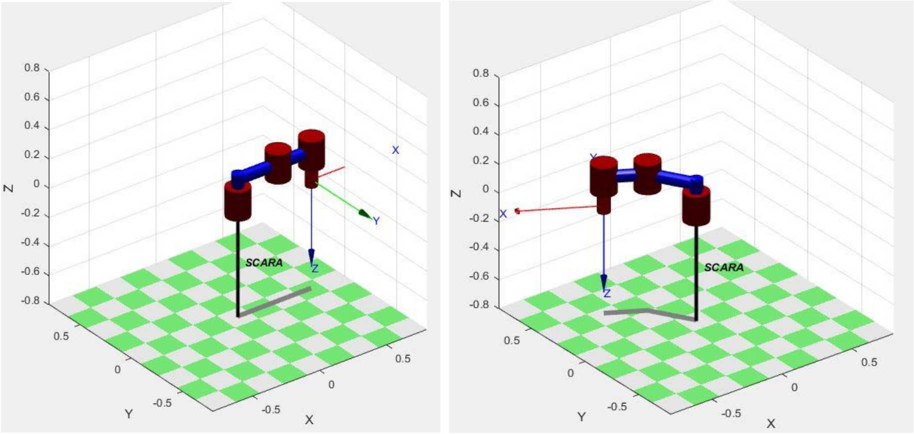


Figure 3.18: Initial pose and final pose respectively.

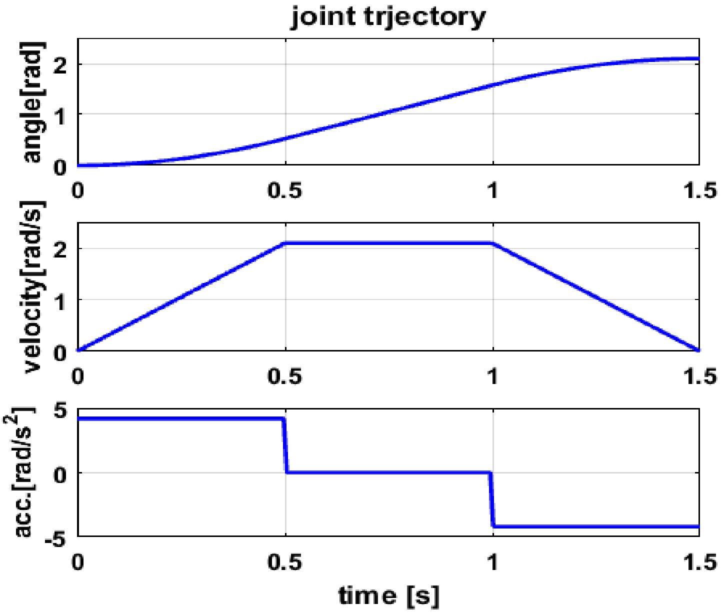


Figure 3.19: Joint 1 (q_1) trajectory.

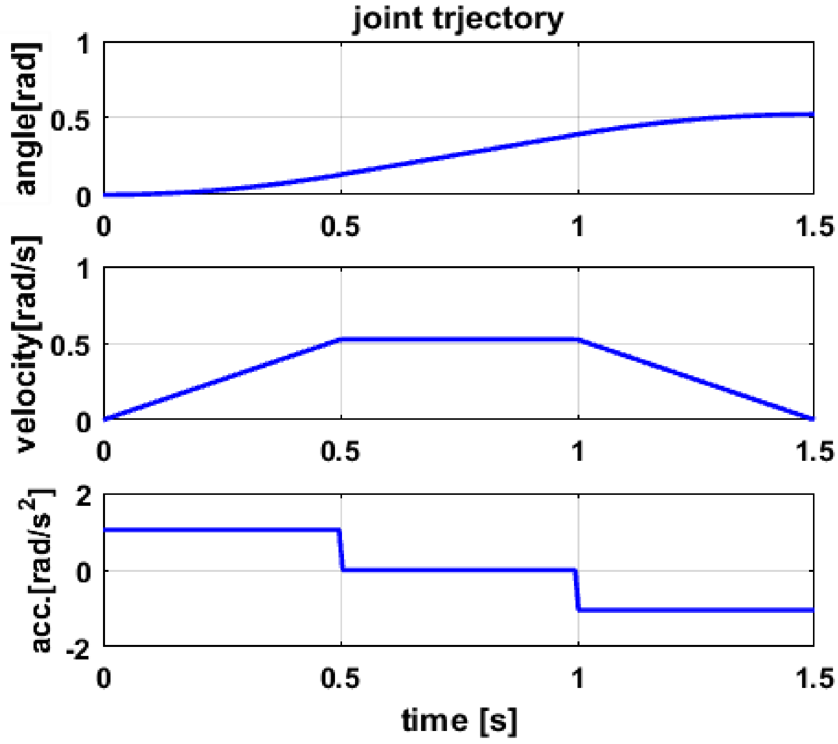


Figure 3.20: Joint 1 ($\boldsymbol{q_1}$) trajectory.

Apply the trajectory of Joint 1 on Eq(46) to assign the first joint torque that would guarantee this trajectory. Also, do so for Joint 2.

Mechanical power for rotational systems is given by:

$$P_i = \tau_i \times \omega_i \tag{50}$$

The desired torques for Joint 1 and Joint 2 to achieve the desired trajectory, also the mechanical power corresponding each joint are shown in Figure 3.21 and Figure 3.22, respectively.

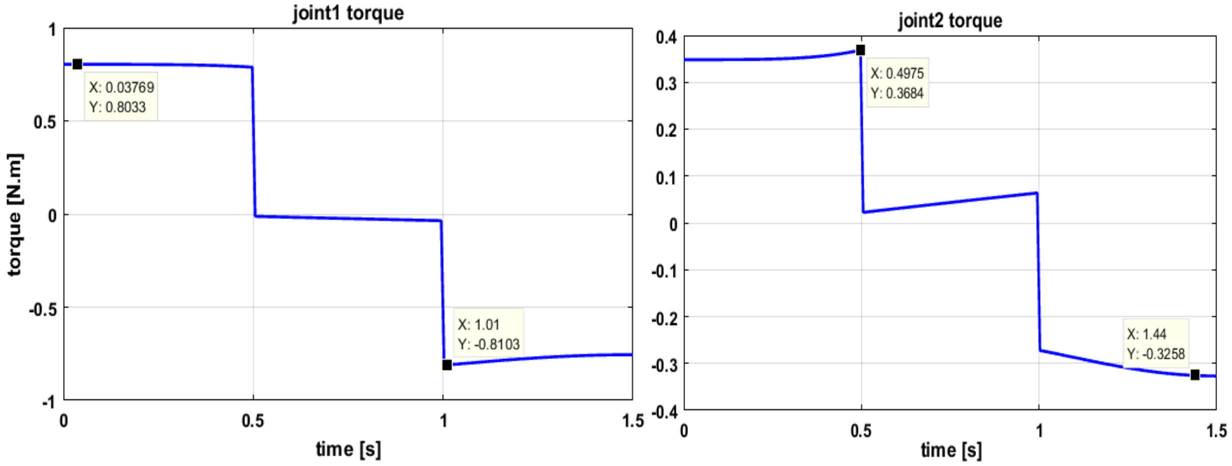


Figure 3.21: Joint 1 torque and Joint 2 torque.

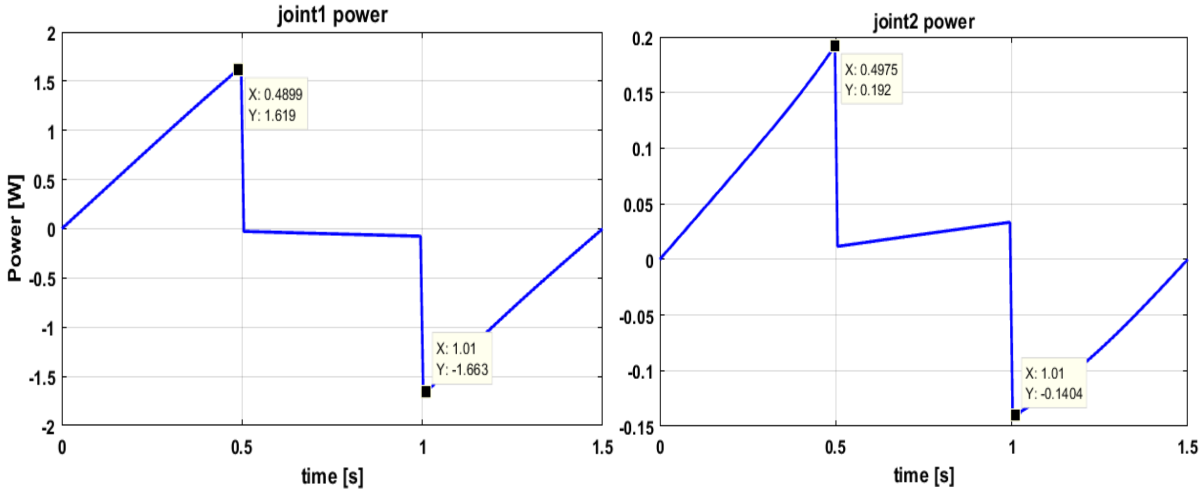


Figure 3.22: Joint 1 power and Joint 2 power.

Here the maximum torque for Joint 1 was 0.81 $N.m$ and for Joint 2 was 0.325 $N.m$. The maximum mechanical power in Joint 1 was at the point where the velocity is maximum and also where the torque is maximum, i.e. at 0.489 s with a magnitude of 1.619 W. The maximum mechanical power for Joint 2 was 0.192 W at 0.49 s (see Figure 3.21 and Figure 3.22).

Second scenario:

The end-effector must move from point $(0.5,0,0)m$ to point $(-0.168,-0.416,0)m$ in 2.2 seconds as shown in Figure 3.23. Where $\vartheta_1$and ϑ_2 are $(0°\ to\ 221.91°)$ and $(0°\ to\ 52.3°)$ respectively. Joint 1 and Joint 2 trajectories are shown in Figure 3.24 and Figure 3.25.

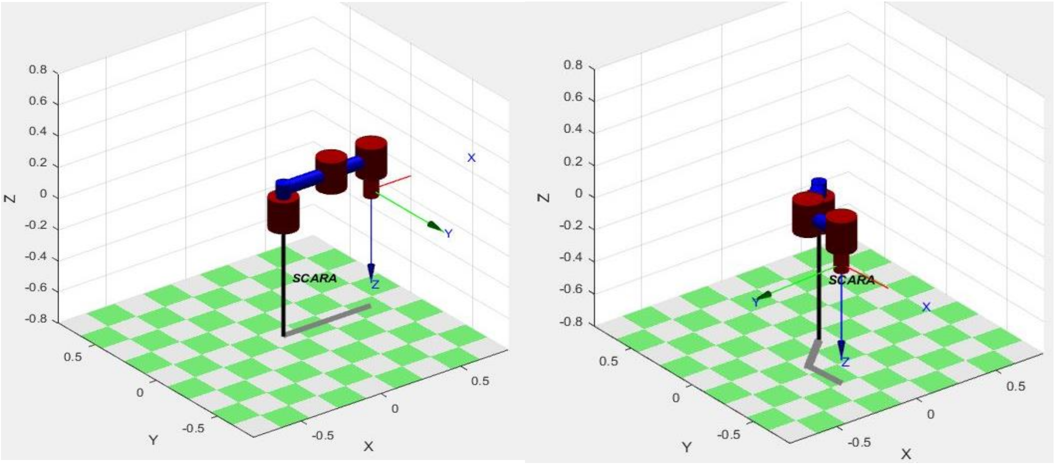


Figure 3.23: Initial pose and final pose respectively.

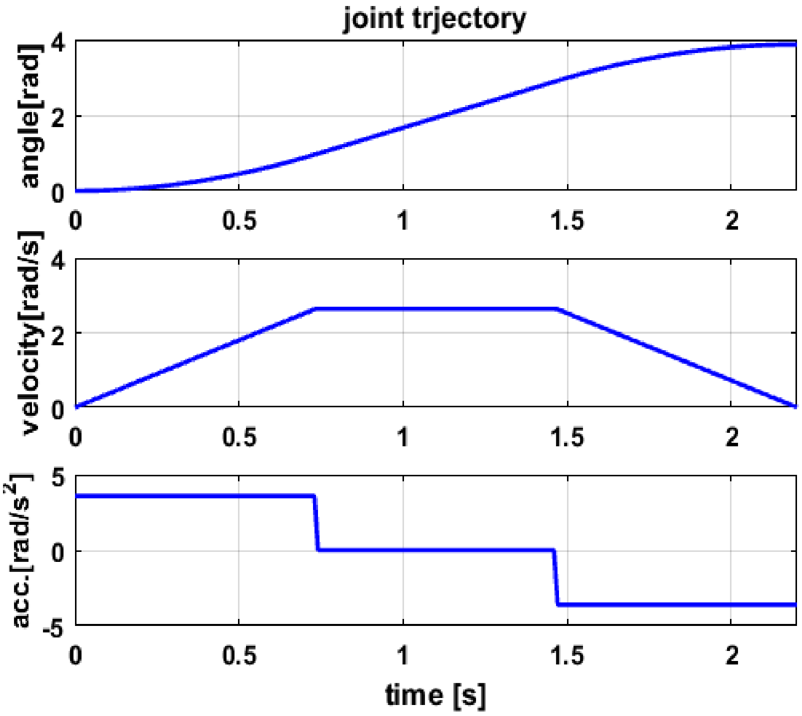


Figure 3.24: Joint 1 $(\boldsymbol{q_1})$ trajectory.

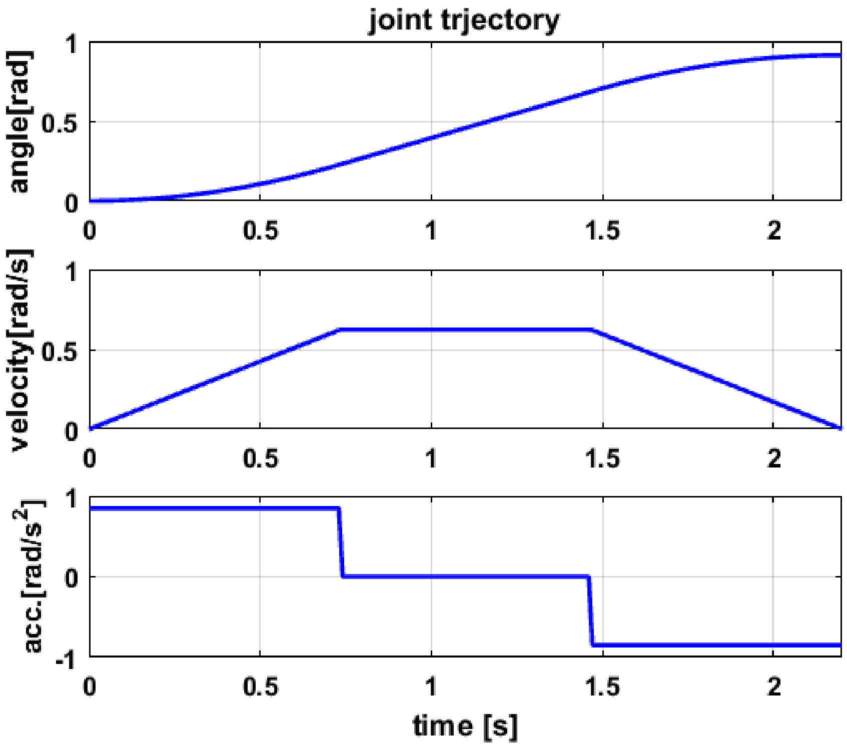


Figure 3.25: Joint 1 (q_1) trajectory.

The maximum torque for Joint 1 was 0.7055 $N.m$ and power of 1.70 W. For Joint 2 the maximum torque was 0.353 $N.m$ and power of 0.218 W (see Figure 3.26 and Figure 3.27).

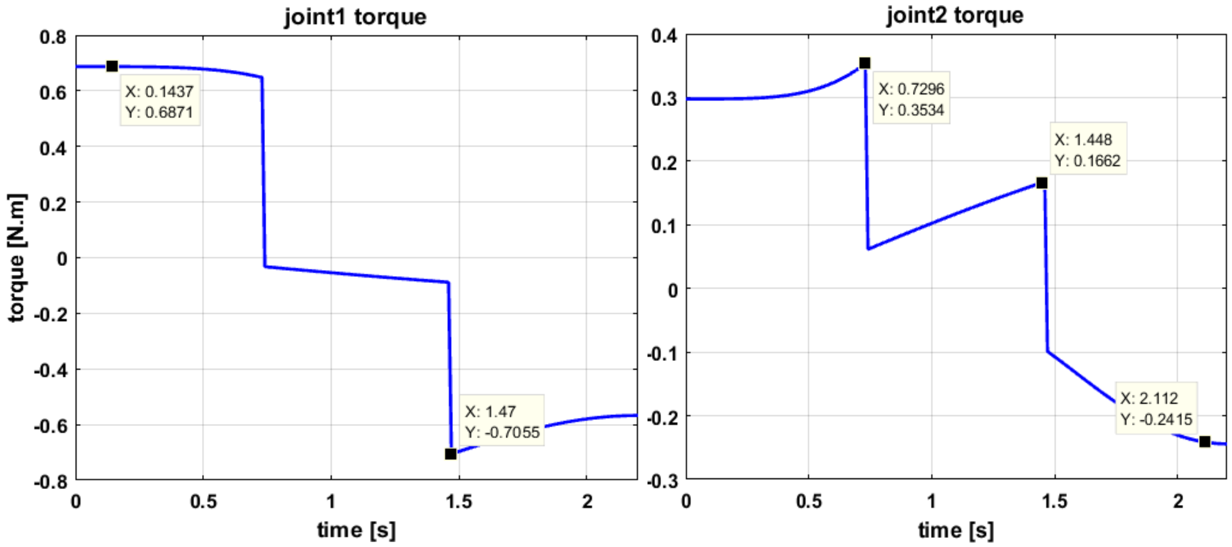


Figure 3.26: Joint 1 torque and Joint 2 torque.

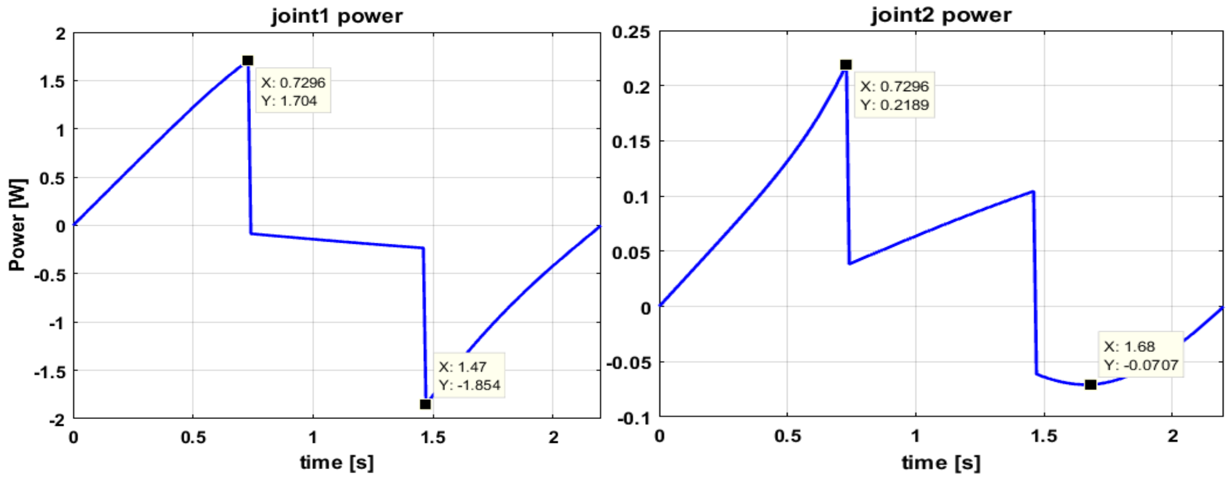


Figure 3.27: Joint 1 power and Joint 2 power.

Third scenario:

The end-effector must move from point $(0.5,0,0)m$ to point $-0.5,0,0)m$ in 1.7 seconds. Where ϑ_1 $(0° \ to \ 180°)$ and $\vartheta_2$will remain 0 as shown in Figure 3.28. Joint 1 trajectory are shown in Figure 3.29.

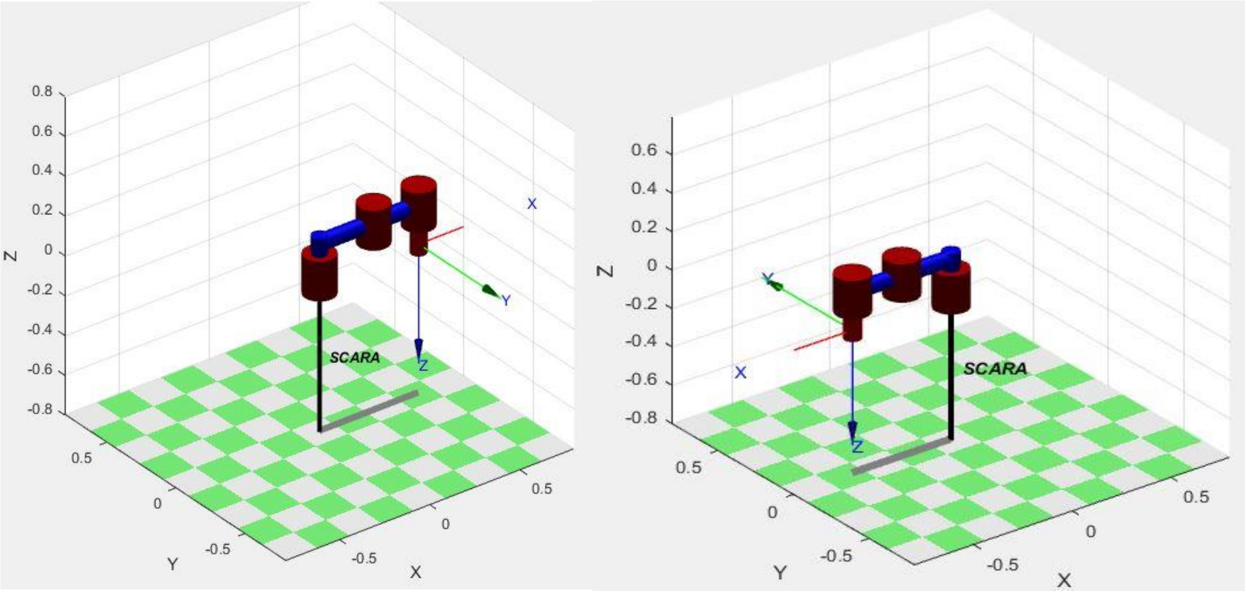


Figure 3.28: Initial pose and final pose respectively.

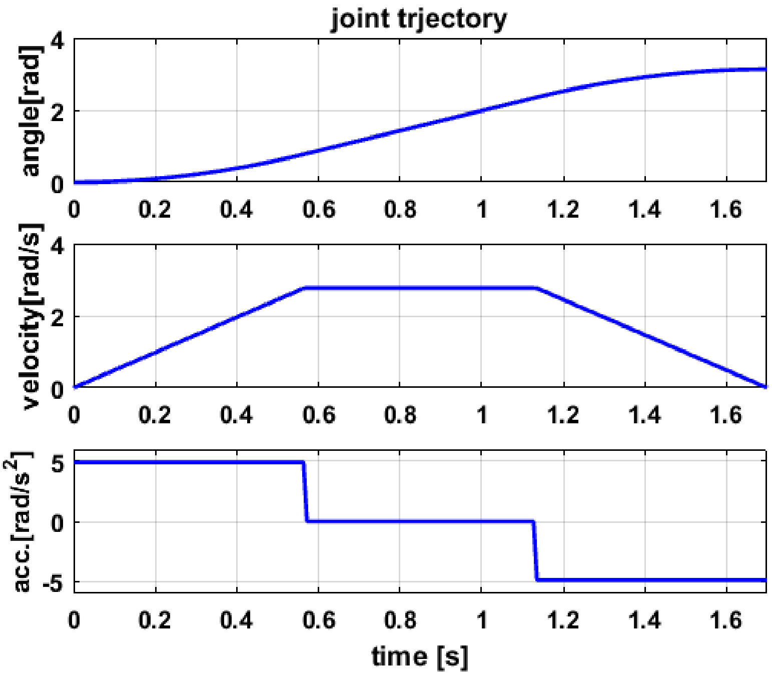


Figure 3.29: Joint 1 (q_1) trajectory.

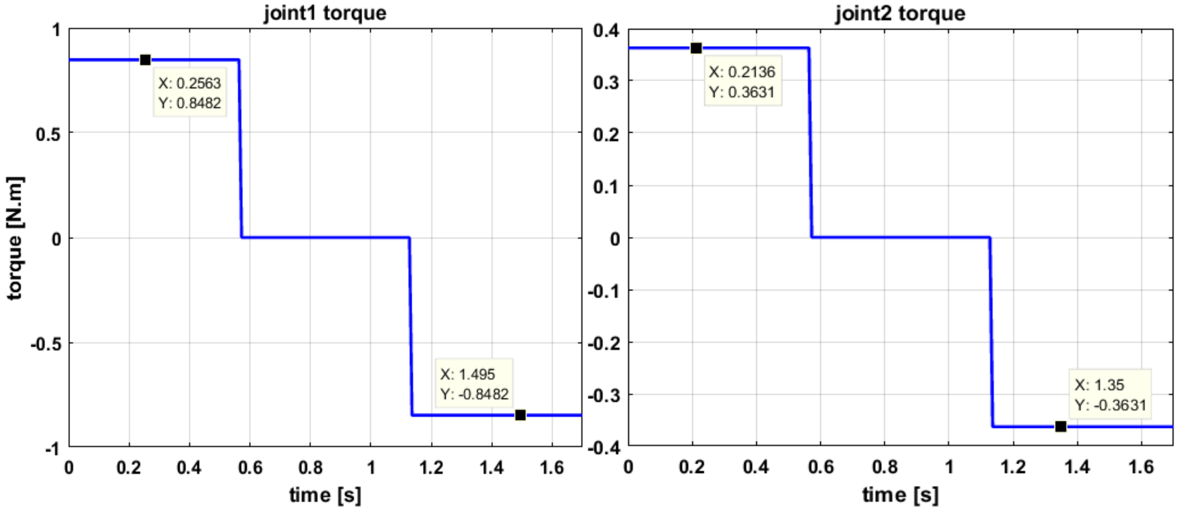


Figure 3.30: Joint 1 torque and Joint 2 torque.

The maximum torque for Joint 1 was 0.848 $N.m$ and for Joint 2 the maximum torque was 0.3631 $N.m$ (see Figure 3.30). This was the worst scenario because it yielded the maximum torque in each joint.

The mechanical power in this scenario shown in Figure 3.31.

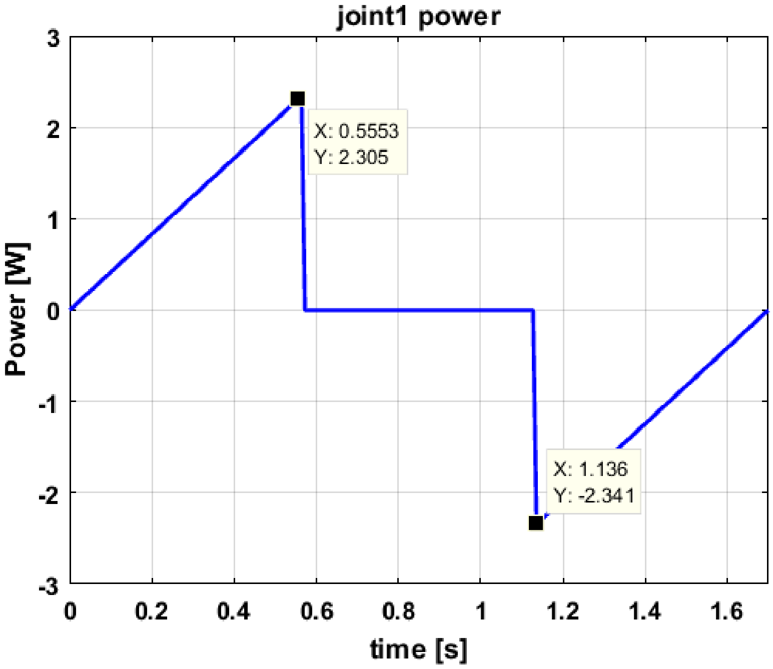


Figure 3.31: Joint 1 power.

Motor sizing summary

The maximum torques yielded from the last scenario, so that the motors must deliver those torques to actuate the manipulator properly. By choosing a factor of safety of 1.5 , the torque for Motor 1 (Joint 1) and for Motor 2 (Joint 2) given by:

$$\tau_1 = 1.272\ N.m$$

$$\tau_2 = 0.5415\ N.m$$

For Joint 3 and Joint 4 a small and cheap servomotor with high torque (0.5 $N.m$) in comparison with its size (see Appendix A).

3.4 Mechanical structure implementation

Robot parts have been machined using plasma CNC machine, Figure 3.32 illustrates the machined parts, and the main stand rod has been machined using lath machine. Then the pulleys for the power transmitters are bored to put the bearings on them. Screws are used for fixturing the rigid parts of the robot links (see Figure 3.33). Figure 3.34, Figure 3.35, Figure 3.36 and Figure 3.37 illustrate the right, top, front and isometric views for the assembled robot's mechanical structure, respectively.

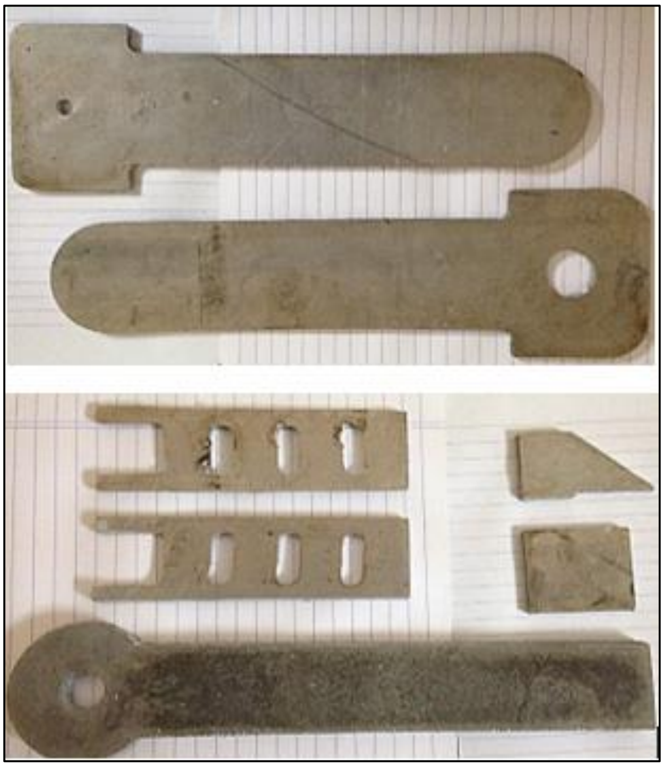


Figure 3 32: The machined parts using Plasma CNC

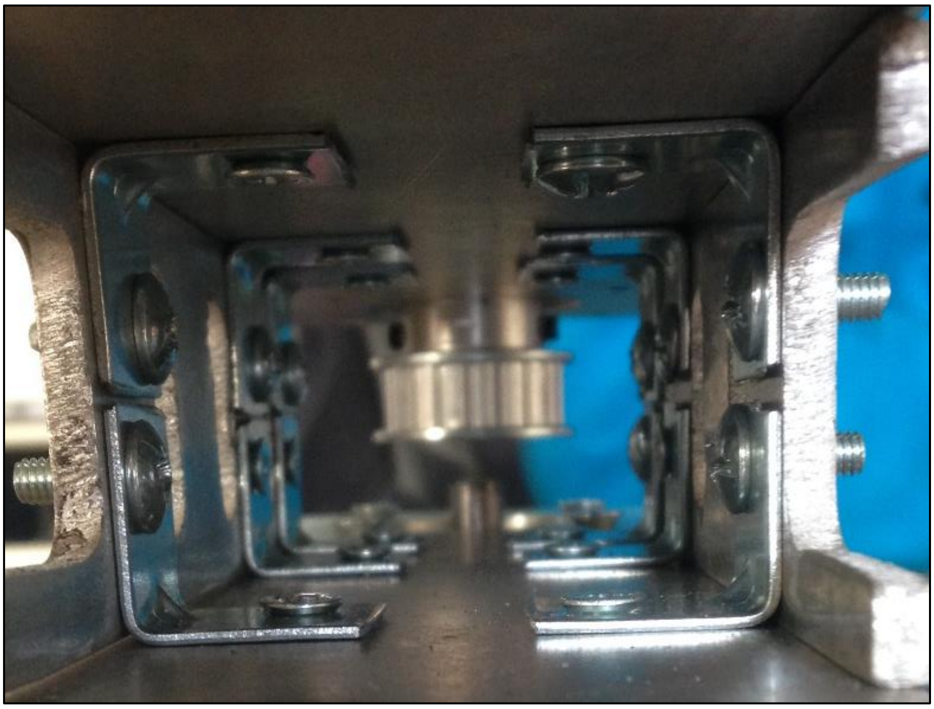


Figure 3.33: First Link fixturing

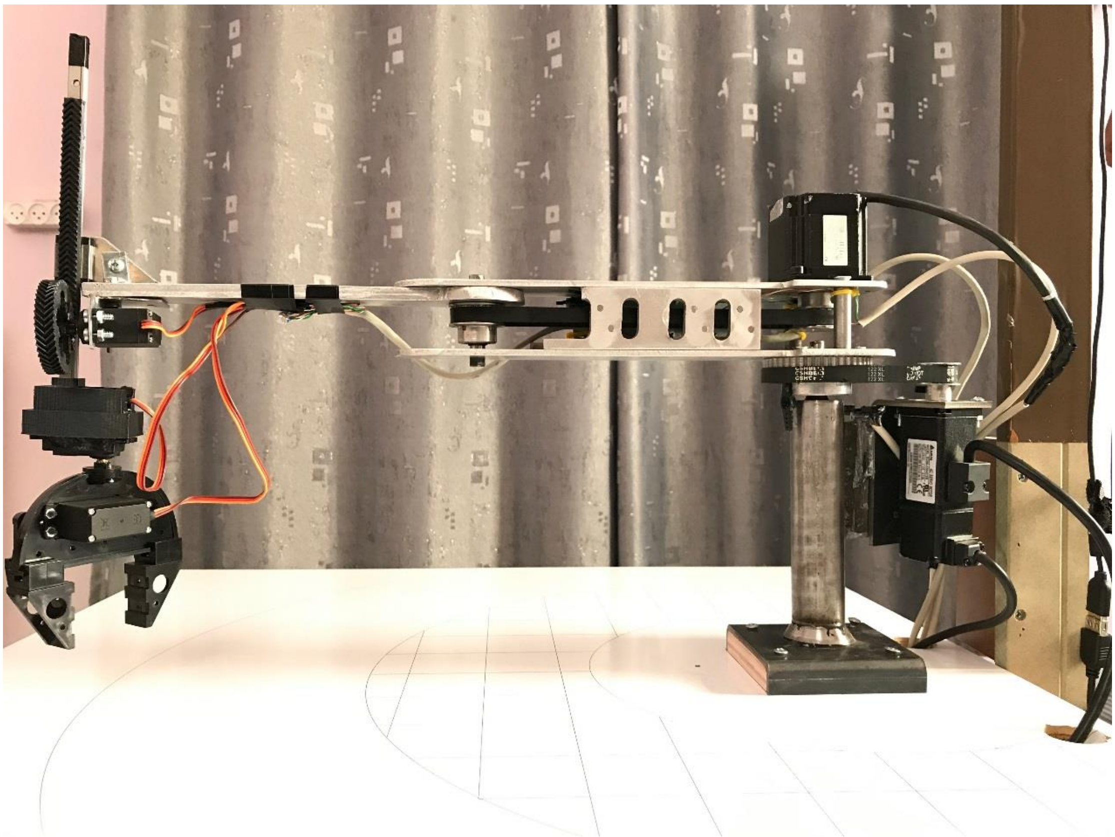


Figure 3.34: Right view of the robot’s mechanical structure

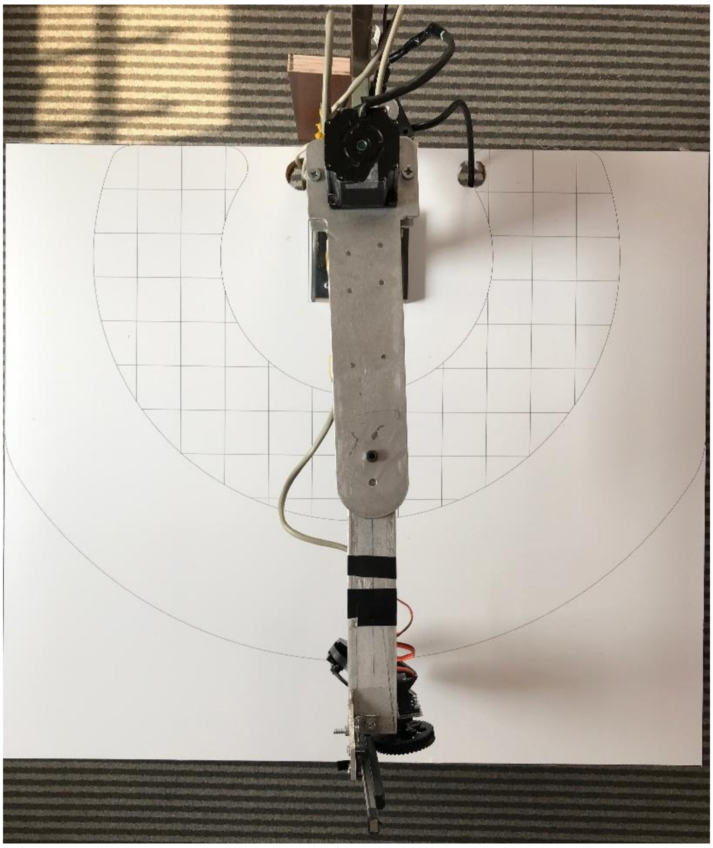


Figure 3.35: Top view of the robot’s mechanical structure

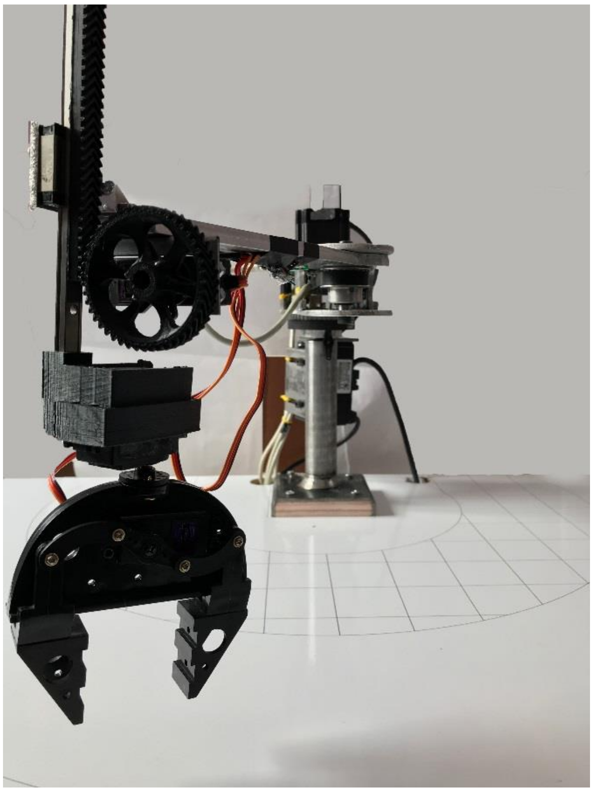


Figure 3.36: Front view of the robot’s mechanical structure*.*

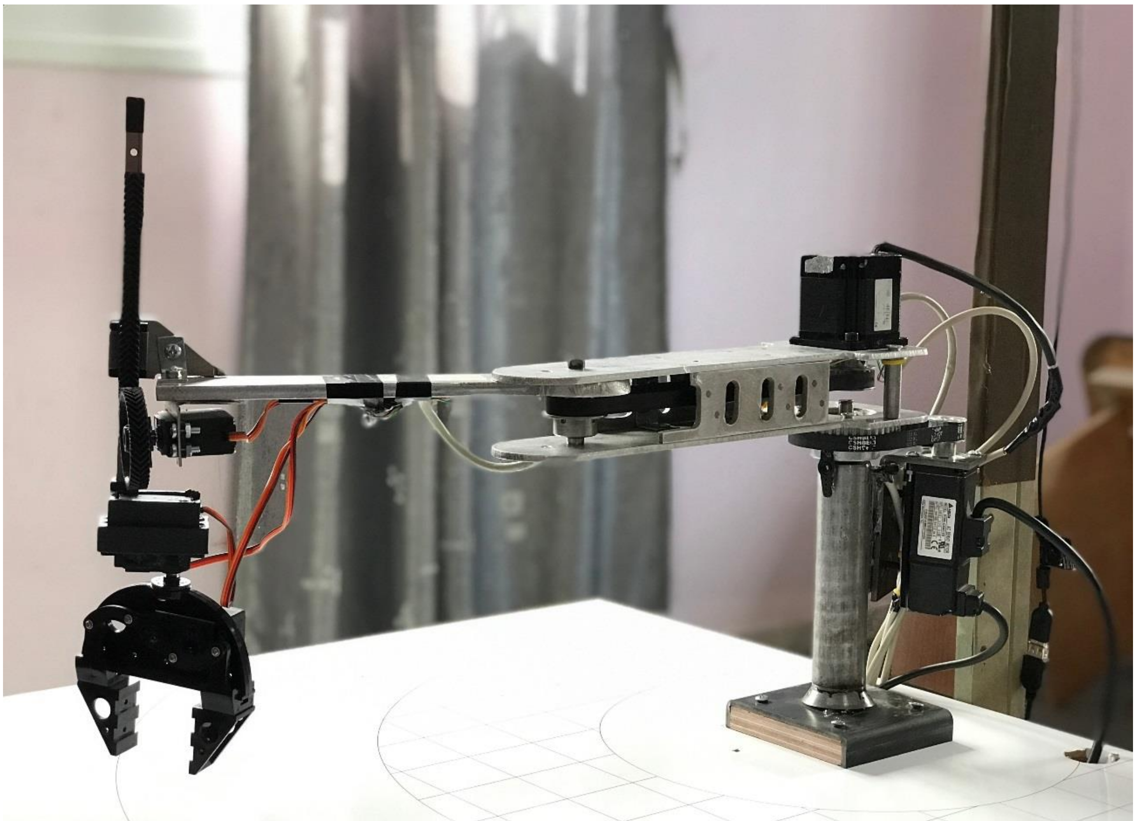


Figure 3.37: Isometric view of the robot’s mechanical structure.

Figure 3.38, shows the right view for the main rod of the robot. The fixture of the servomotor is implemented to be adjustable to ensure the tension in the timing belt.

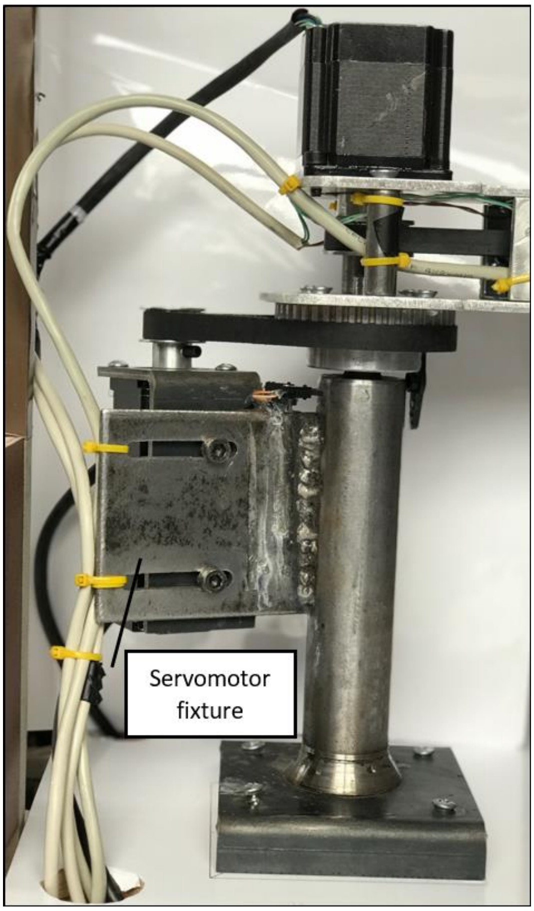


Figure 3.38: The main rod for the robot.

The mechanical structure shown a good dynamic performance, with smooth movement on the revolute joints. A small elastic deformation occurs in the farthest point of the arm, where a very small vibration occurs when moving the Third joint up or down but these vibrations do not affect very much the arm to accomplish its mission. However, the structure need farther enhancements and optimizations for a mission that needs a very accurate positioning or sensitive manipulation.

4

Chapter 4

Electrical design

4.1 Electrical circuits

A very important part of this project, is the electrical circuits, they are used to send and receive the signals from motors drivers, limit switches and encoder.

The interfacing circuit between the MyRio-1900 controller and the servo motor driver, the stepper motor driver and the noisy encoder signals is shown in Figure 4.1.

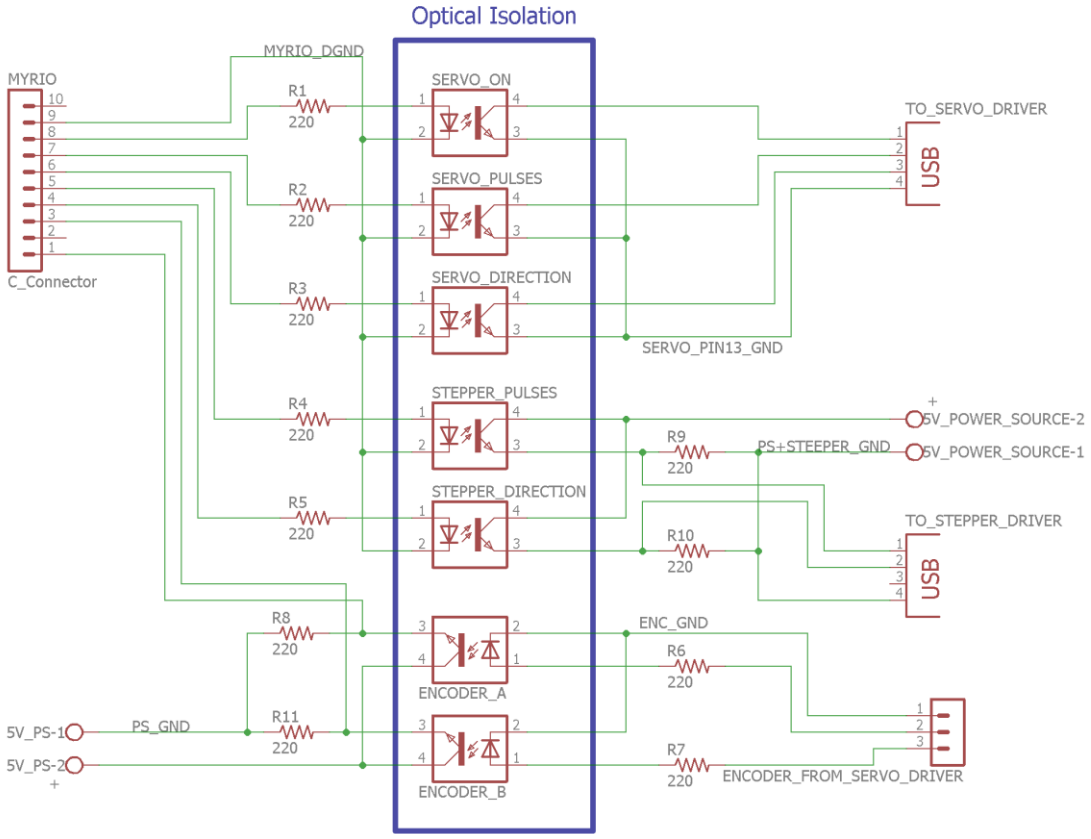


Figure 4.1: Optical isolation circuit.

As it is known, the servo motor driver makes a lot of noise on the wirings due to its electromagnetic field, so the optical isolation must be used to separate grounds, filter the noisy signals and to get smooth readable pulses from the encoder.

The limit switches in this project are very important, they are used to know the exact position of each link and to limit the movement range of the links, the only way to get rid of them is to get an absolute encoder for each motor, which is the expensive way.

Figure 4.2 shows the electrical circuit that used to get a digital signal of each condition of the limit switch.

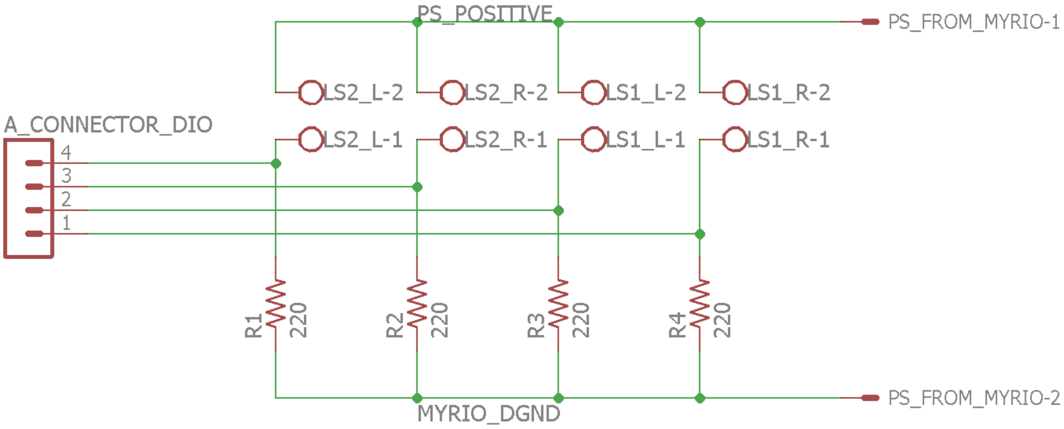


Figure 4.2: Limit switches interfacing circuit.

The pull-down resistors are used to prevent the short circuit from happening when a limit switch is pressed.

The third joint, fourth joint and the gripper mechanism are actuated using small servomotors, where these motors are supplied by external DC-power supply with (1.5 A) maximum current. Figure 4.3 illustrates the electrical circuit for these motors.

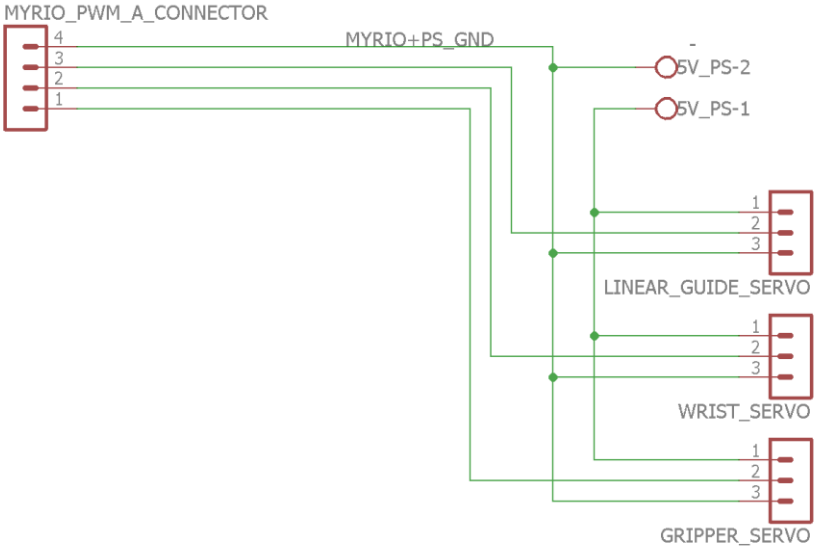


Figure 4.3: circuits for the servomotors of the third joint, fourth joint and gripper.

4.2 Electrical components

In this section a detailed description is presented for each component mentioned in the electrical design.

4.2.1 Motors

The main factors for choosing the robot actuators are the availability of them in the market and the motor rated torque and speed to ensure that they are able to actuate the robot mechanisms as illustrated in Section 3.3.

1- AC-Servo motor

A Delta-servomotor has chosen as an actuator for the first joint (see Figure 4.4), with the following characteristics:

- Voltage: $220\ v$
- Maximum current: $2.7\ A$
- Rated speed: $3000\ rpm$
- Power output: $100\ W$
- Rated torque: $0.32\ N.m$
- Weight: $0.5\ Kg$

Figure 4.4: Delta-servomotor.

To obtain 1.2 $N.m$, which is the required torque for the first joint two pulleys with $1{:}\,4$ gear ratio is used for the power transmitter to actuate the first joint.

2- Stepper Motor

The principle of stepper motor, that the rotor rotates in discrete mechanical steps. A change in phase current from one state to another creates a single step change in the rotor position. If the phase current state is not changed, the rotor position stays in that stable position [13]. Position control in the stepper motor is in open loop fashion (see Figure 4.5).

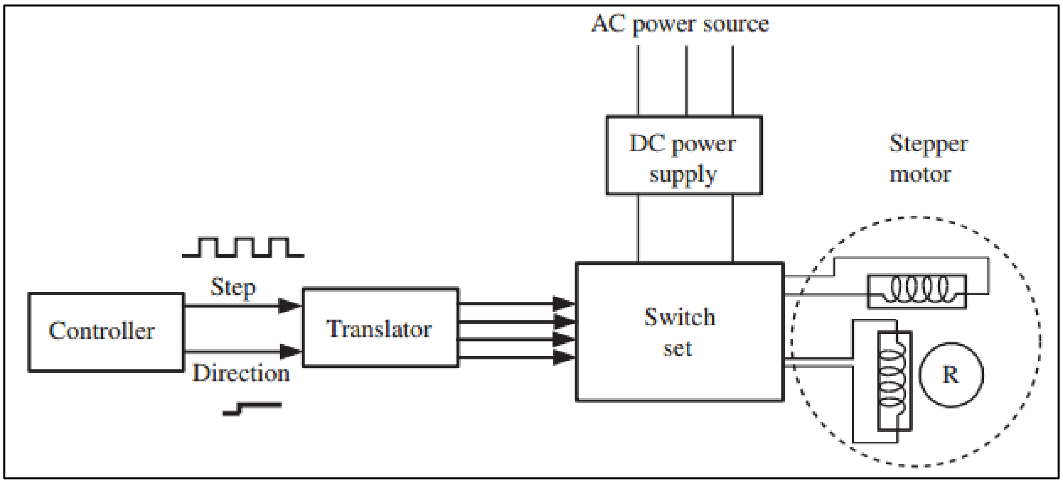


Figure 4.5: Stepper motor control block diagram [13].

The advantages of stepper motors are their low coast, simplicity of design, ruggedness and simplicity of operation. The drawbacks of using stepper motors are the motor may slip for one step or more, where this will affect the output position where no feedback from the output and the accuracy is dependent to the rotor mechanical structure [13]. The stepper motor is used to actuate the Second joint.

3- DC servo motor high torque " MG 996R"

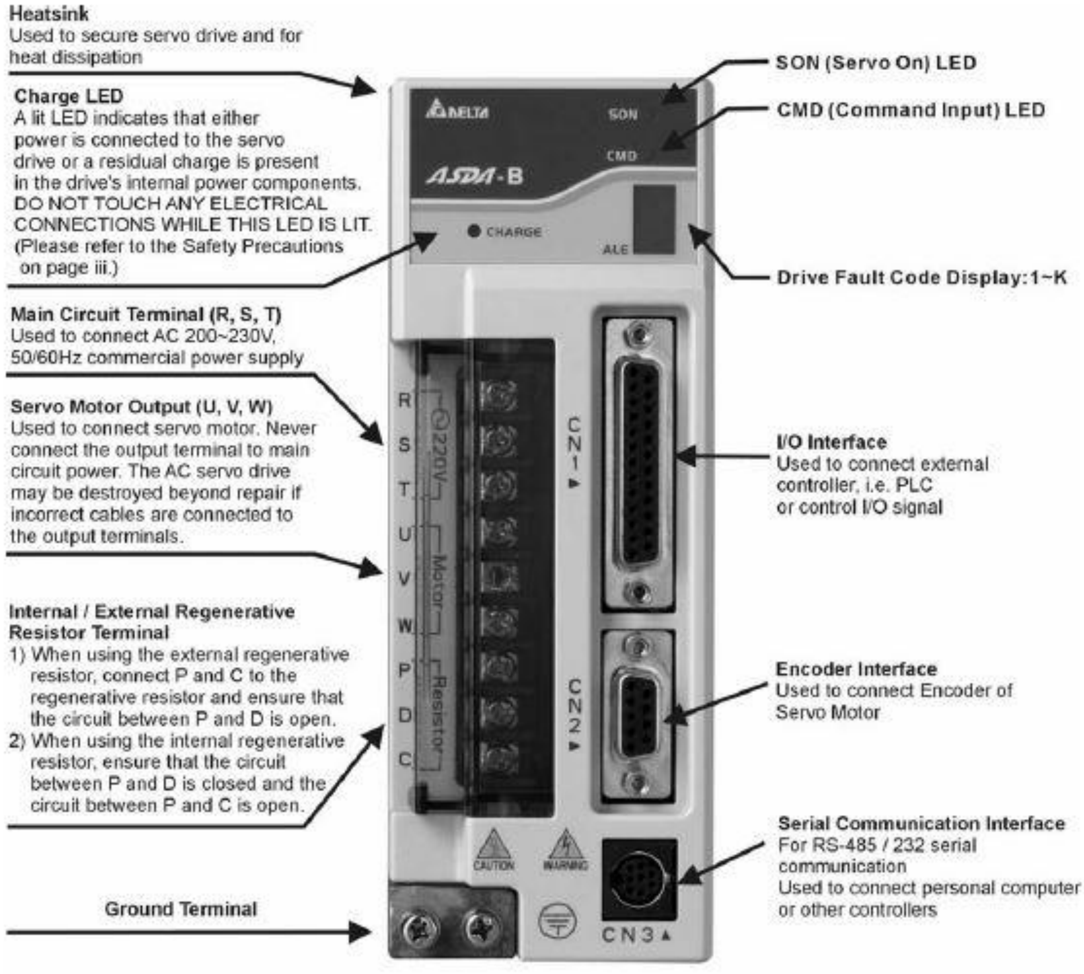
This motor (see Figure 4.6) used in End effector mechanisms (for Rack and pinion, wrist joint, and gripper) with the following characteristics:

- Weight: 55 g
- Operating voltage: 4.8 V to 7.2 V
- Stall torque: 9.4 kg·f·cm (4.8 V), 11 kg·f·cm (6 V)
- Stall Current: 2.5 A
- Rotate angle: 180 degrees.

Figure 4.6: servo motor MG996R.

4.2.2 Drivers

1- **Servo motor Drive**



Heatsink
Used to secure servo drive and for heat dissipation

Charge LED
A lit LED indicates that either power is connected to the servo drive or a residual charge is present in the drive's internal power components. DO NOT TOUCH ANY ELECTRICAL CONNECTIONS WHILE THIS LED IS LIT. (Please refer to the Safety Precautions on page iii.)

Main Circuit Terminal (R, S, T)
Used to connect AC 200~230V, 50/60Hz commercial power supply

Servo Motor Output (U, V, W)
Used to connect servo motor. Never connect the output terminal to main circuit power. The AC servo drive may be destroyed beyond repair if incorrect cables are connected to the output terminals.

Internal / External Regenerative Resistor Terminal
1) When using the external regenerative resistor, connect P and C to the regenerative resistor and ensure that the circuit between P and D is open.
2) When using the internal regenerative resistor, ensure that the circuit between P and D is closed and the circuit between P and C is open.

Ground Terminal

SON (Servo On) LED

CMD (Command Input) LED

Drive Fault Code Display:1~K

I/O Interface
Used to connect external controller, i.e. PLC or control I/O signal

Encoder Interface
Used to connect Encoder of Servo Motor

Serial Communication Interface
For RS-485 / 232 serial communication
Used to connect personal computer or other controllers

Figure 4.7: Servo driver features [14].

The power of driver (see Figure 4.7) is single phase, and it convert the single phase to three phases to supply the motor. The main circuit terminal is used to supply the servo with line power. If a single-phase supply, is used connect the R and S terminals to power. If three phase, connect all three R, S, and T terminals.

And (U, V, W) is Used to connect servo motor with driver, and the encoder interface connect directly with motor and the I/O interface used to connect with controller.

And the serial communication interface used to interface the computer with driver program [14].

Basic Wiring Schematic of 100W models is shown in Figure 4.8.

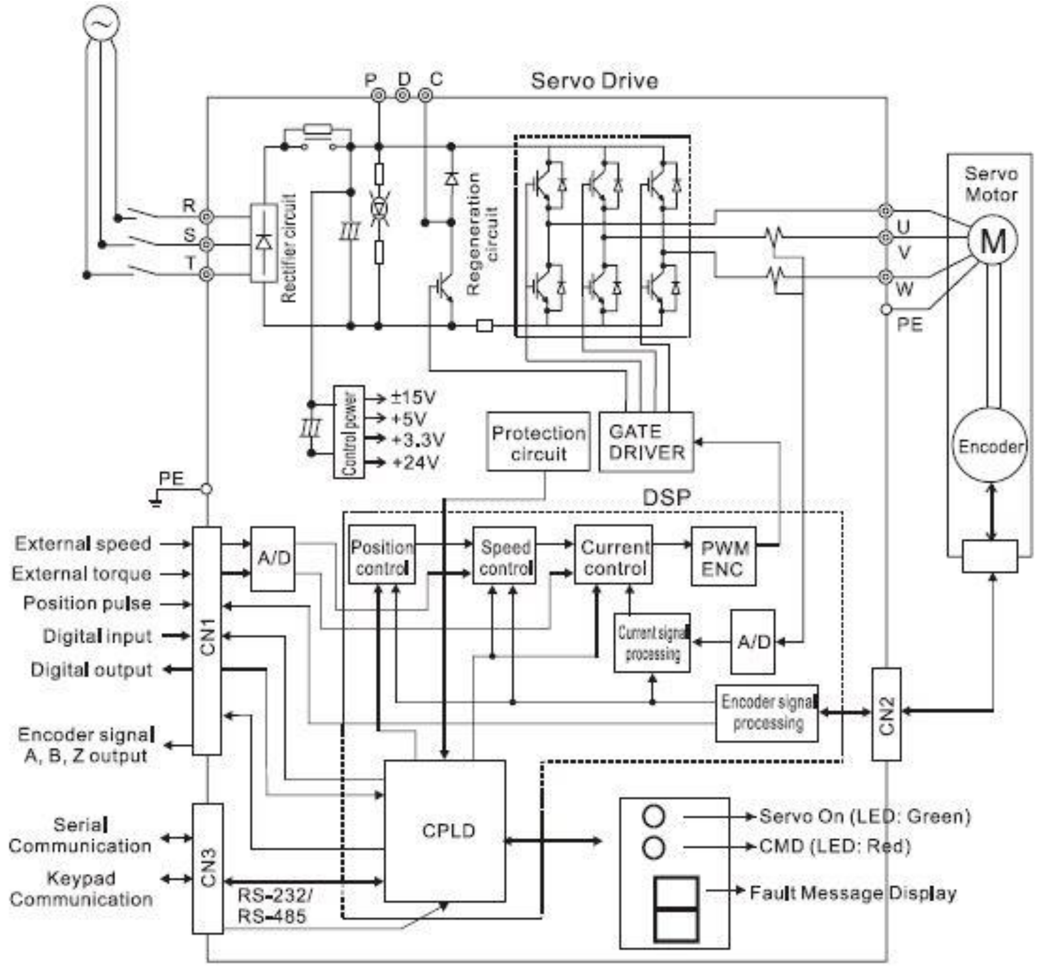


Figure 4.8: Schematic diagram of AC servo motor drive [14].

Figure 4.8 represents the internal component of a driver. The Delta servomotor in this project will be closed loop with its driver (see Figure 4.9), and open loop with computer.

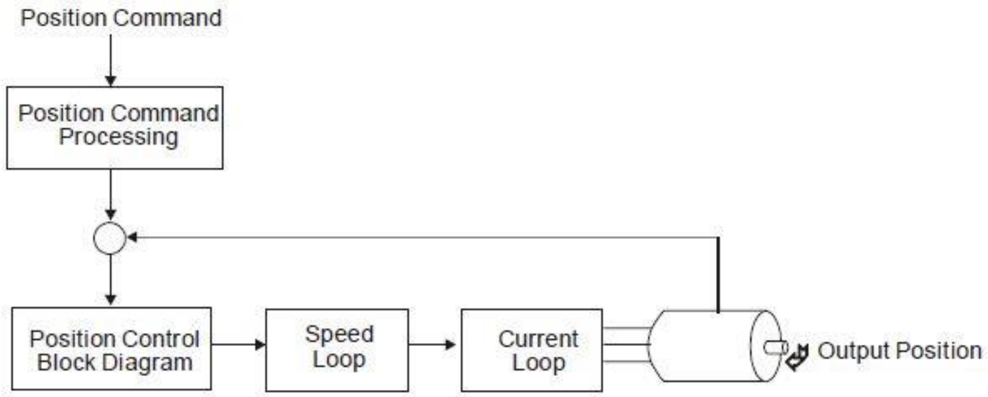


Figure 4.9: Closed loop position control.

2- **Stepper Motor Drive**

HY-DIV268N-5A stepper motor drive model will used in the project, and the specification of the driver as follows (see Appendix B):

- DC 12 ~ 48V power supply.
- Output current of 0.2A ~ 5A.
- Weight 200 grams.
- high starting speed.
  high-speed torque.

4.2.3 MyRio-1900 Controller

All systems in this project integrated with each other and programmed using high level pregaming language under LabView software with MyRio-1900 (see Figure 4.10), where it provides (ADC,DAC,DAQ and Digital I/Os) [15], with good characteristics that allow the systems to integrate with each other in real-time.

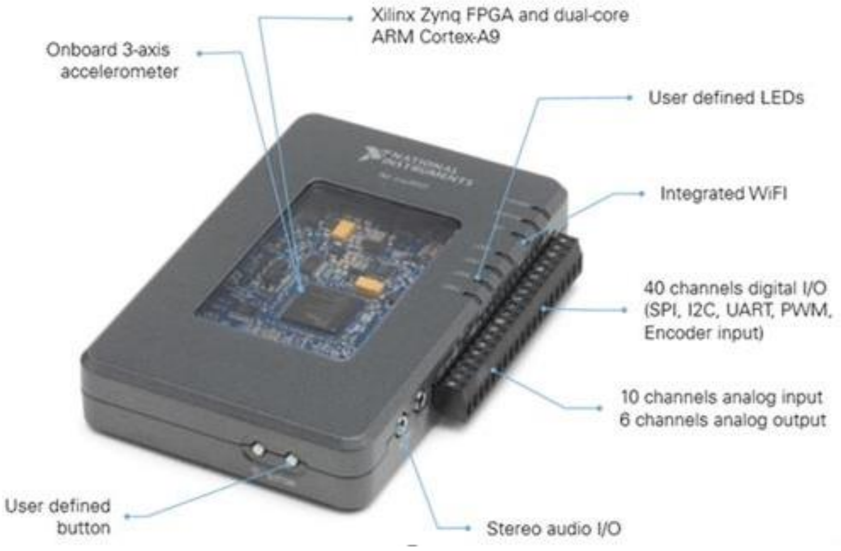


Figure 4.10: My-Rio Controller [15].

The controller has forty digital input/output, some of there for pulse width modulation and others for Encoder, and it have a ten channels analog input and six channels analog output [15]. MyRio-1900 peripherals block diagram is detailed in Figure 4.11.

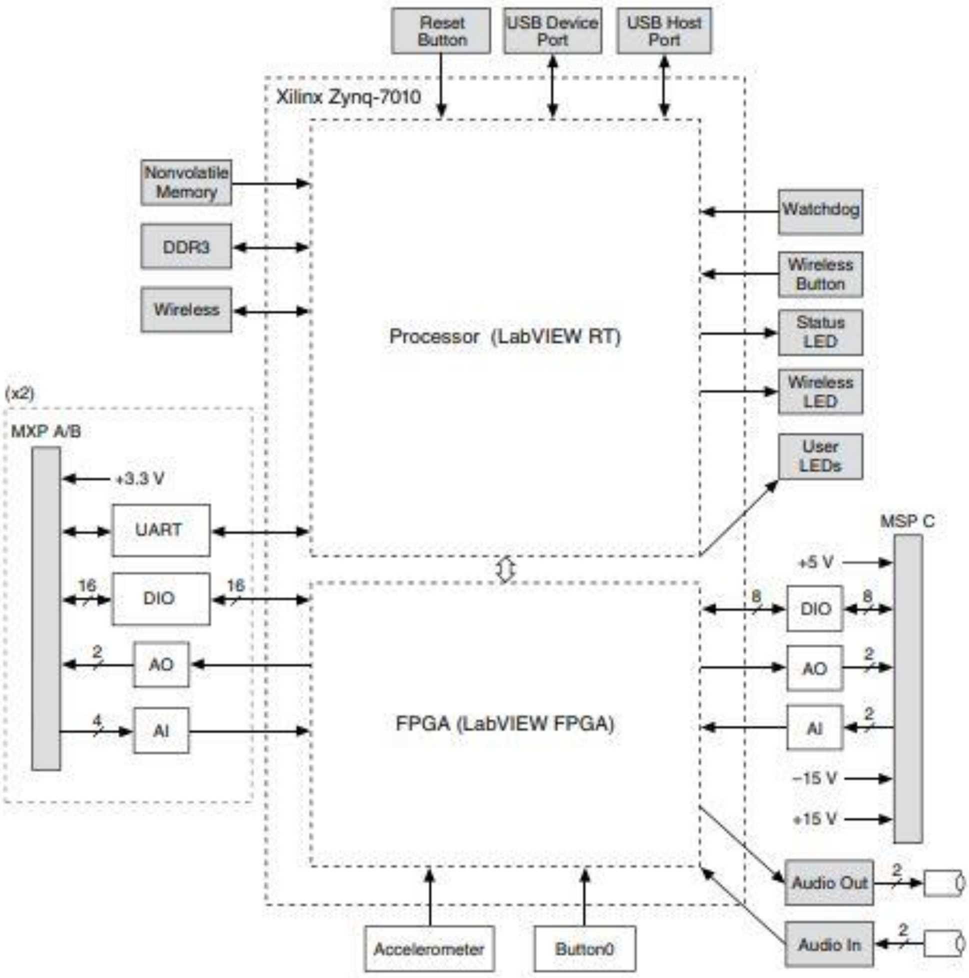


Figure 4.11: MyRio-1900 Hardware block diagram [15].

4.2.4 Measurement systems

1- HD Camera

HD webcam C270 used in this system, the specification are as follows:

- HD video capture: up to 1280 × 720 pixels.
- Photos: up to three megapixels.
- Power: USB 2.0 from computer.

Figure 4.12: HD Camera.

2- Limit Switch

Limit switches are used for robot homing process also for safe manipulation.

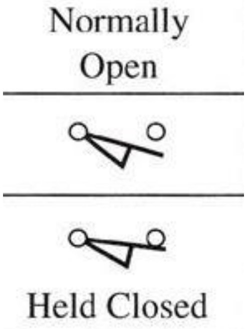


Figure 4.13: Limit switch symbol

4.3 Electrical circuits implementation

Figure 4.14, Figure 4.15 and Figure 4.16 show the implemented optical isolation circuit, Limit switches interfacing circuit and the interfacing circuit between the MyRio-1900 with the stepper motor driver and the optical encoder, respectively.

The optical isolation circuit was machined using PCB CNC machine, then welded by hand. The other circuits were implemented by welding the connections on a ready-made PCB.

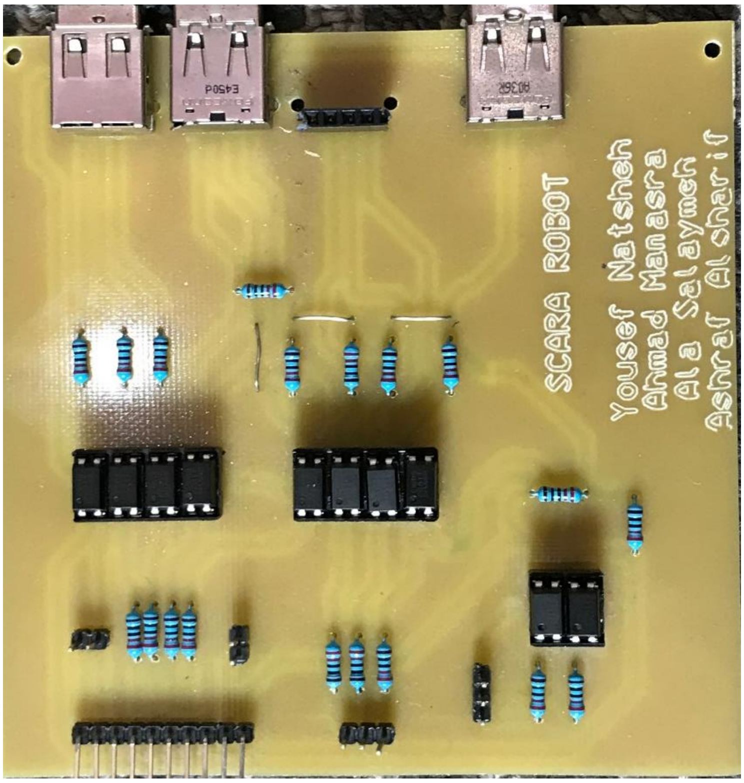


Figure 4.14: Optical isolation circuit between MyRio-1900 and servomotor interfacing circuit

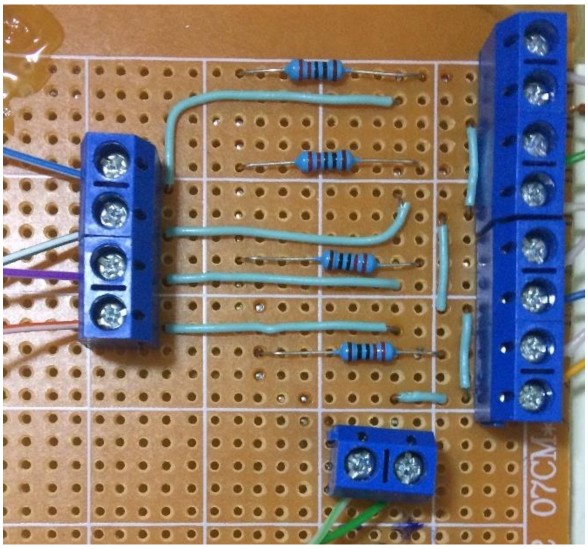


Figure 4.15: Limit switches interfacing circuit.

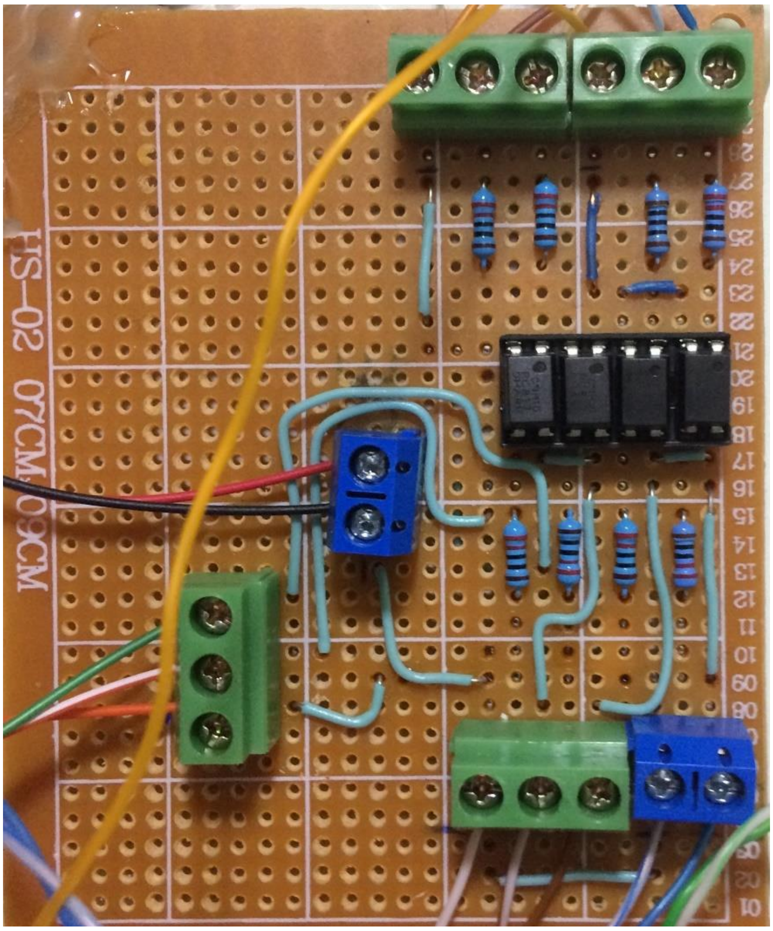


Figure 4.16: Interfacing circuit for the stepper motor driver and the optical encoder.

5

Chapter 5

Vision System and image processing

One of the main components of the robots that are dedicated for sorting missions, is the vision system, it contains at least one camera connected to a computer as shown in Figure 5.1 [16]. The vision system can recognize the color, shape and the poses of the objects.

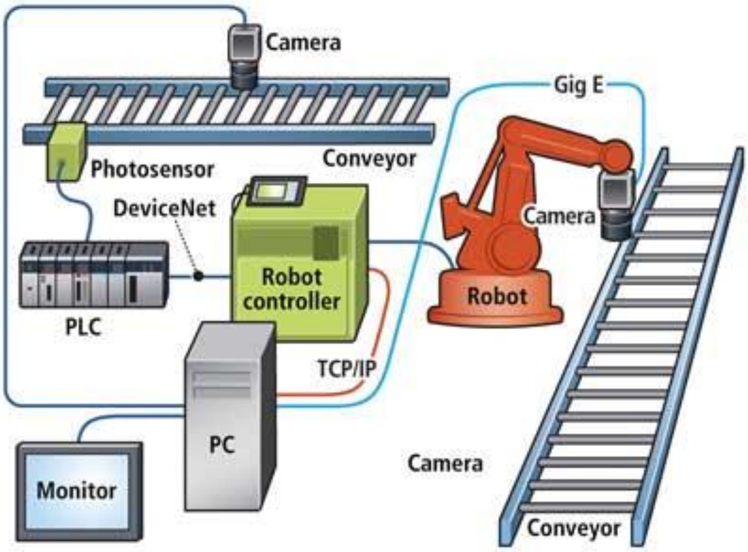


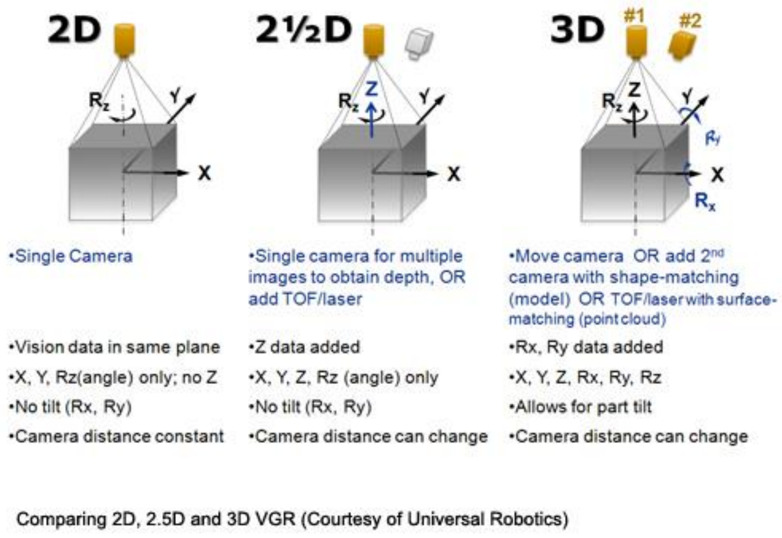
Figure 5.1: Camera connection with the computer [16].

5.1 The vision-based measurements in this project

For each object in the workspace, the color and pose data will be extracted by processing the images that are acquired from the digital camera. The visual data from the digital camera requires complex transformations to extract numerical-feasible information about the objects in the workspace [8], where in this project the pose and color are the desired data needed to be extracted from the vision system. This information will be used to generate the trajectory that the controller of the manipulator has to track.

The vision process in this robotics system is considered as high-level vision, where the recognizing of the objects will be as comparing the objects with a stored objects.

The vision system can even measure the length, height and the depth of the objects using two cameras as shown in Figure 5.2. The robotics application in this project needs a single camera (2D configuration), where the desired measurement is the pose of the object (x, y, θ_z).



2D	2½D	3D
•Single Camera	•Single camera for multiple images to obtain depth, OR add TOF/laser	•Move camera OR add 2nd camera with shape-matching (model) OR TOF/laser with surface-matching (point cloud)
•Vision data in same plane	•Z data added	•Rx, Ry data added
•X, Y, Rz(angle) only; no Z	•X, Y, Z, Rz (angle) only	•X, Y, Z, Rx, Ry, Rz
•No tilt (Rx, Ry)	•No tilt (Rx, Ry)	•Allows for part tilt
•Camera distance constant	•Camera distance can change	•Camera distance can change

Comparing 2D, 2.5D and 3D VGR (Courtesy of Universal Robotics)

Figure 5.2: Measurements from one camera or two cameras [17].

The task of the camera is to measure the intensity of the light reflected by the object, Figure 5.3 shows the camera subsystems. A photosensitive element, called pixel is used to measure the light intensity, where it is capable to transfer the light energy into electric energy. An array of pixels is deployed in image plane, a lens is used for focusing the light reflected from the surrounding environment to the image plane [8].

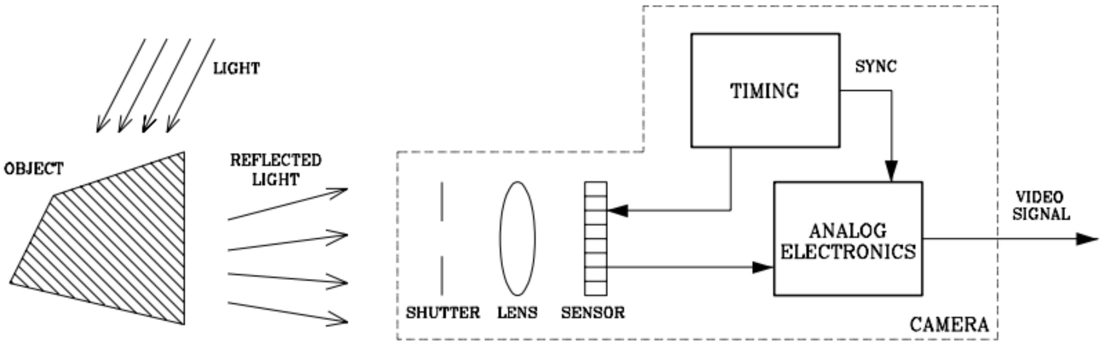


Figure 5.3: Schematic diagram of a vision system [8].

Figure 5.4 illustrates the frames that attached on the image plane, the center of the lens and the base frame of the robot. It is necessary to define the homogenous transformation between the base frame to map the measurements from the image plane and the metric unit measurements.

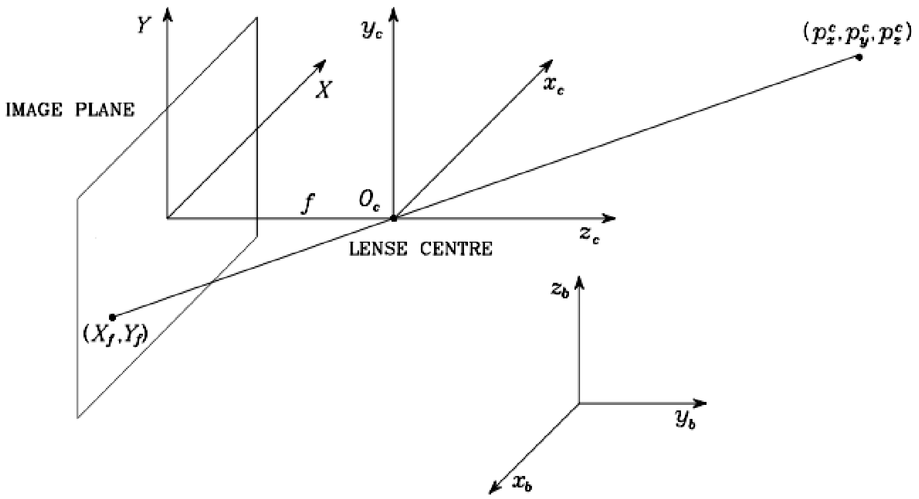


Figure 5.4: Image plane, lens center and base frame [8].

Referring to figure 5.4 consider frame (XY) to be the image plane, frame $C - (x_c\ y_c\ z_c)$ to be the center of the lens frame and frame $b - (x_b\ y_b\ z_b)$ to be the base frame of the robot [8], where the desired measurements in this project needed to be described with respect to the base frame. The video signal from the camera will be in the image plane, so that the transformations between the frames must be well described to yield measurements in metric units to allow the computer, which generate the trajectory for the robot to estimate the poses of the objects with an accepted accuracy.

Consider point $p^c = [p_x^c\ p_y^c\ p_z^c]^T$ that is showed in figure 5.4 to be a point in the space that is described in frame $C - (x_c\ y_c\ z_c)$ and the homogenous transformation between the frame $b - (x_b\ y_b\ z_b)$ and frame $C - (x_c\ y_c\ z_c)$ is T_b^c, hence point p^c is given by:

$$p^c = T_b^c\ p$$

where p is described with respect to the base frame.

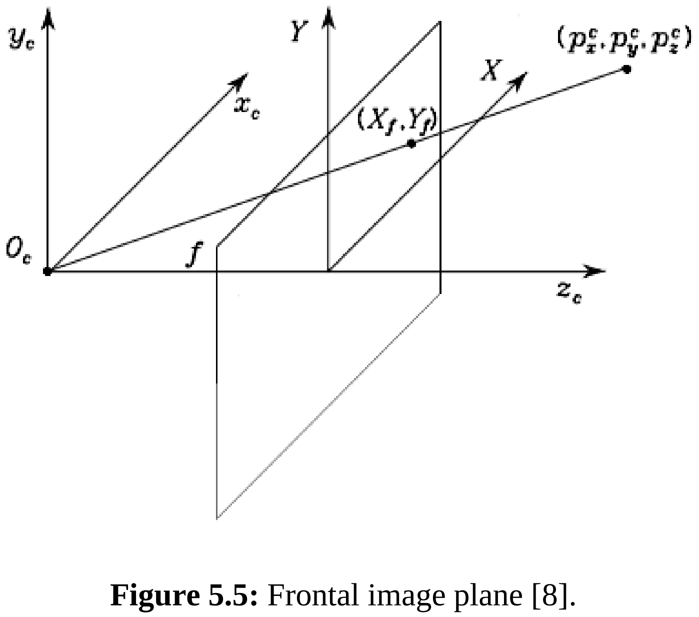


Figure 5.5: Frontal image plane [8].

The overall transformation from the work space of the robot of an observed object to the frontal image plane (see Figure 5.5) is given by the so-called camera calibration matrix, which is given by [8]:

$$\Xi = \Omega \, \Pi \, T_b^c$$

where

$$\Omega = \begin{bmatrix} f\,\alpha_x & 0 & X_0 \\ 0 & f\,\alpha_y & Y_0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

where f is the focal length of the lens, (α_x, α_y) are scale factors that relates between the pixel coordinate and metric coordinate, and (X_0, Y_0) are the offsets which take into account the position of the origin of the pixel coordinate system with respect to the optical axis [8].

In this project, a lighting source will be placed on top of the workspace, to ensure the data to be correct, the camera will be fixed (eye-to-hand) where the camera has a fixed pose with respect to the base frame camera.

The advantage of using the eye-to-hand configuration is that the camera's pose does not change during the execution of the task, this implies that the accuracy of such measurement is constant [8].

5.2 Programming environments

Vision system can be programmed in several environments, some of them are OpenCV, LABVIEW and MATLAB/Simulink.

The vision system is advisable to be built using LabView software due to its systematic programing procedure with high level programing language. The vision system under LabView is compatible with MyRio-1900, this leads to avoid the problems of synchronization between the interfacing circuits and processers between the LabView software and the hardware of the project.

5.3 Vision system programming under LabView

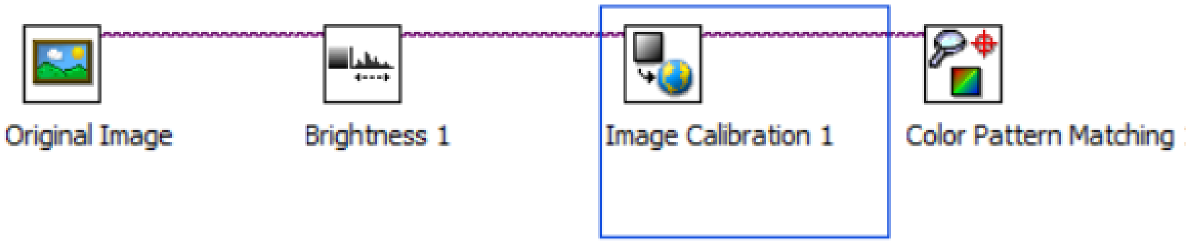
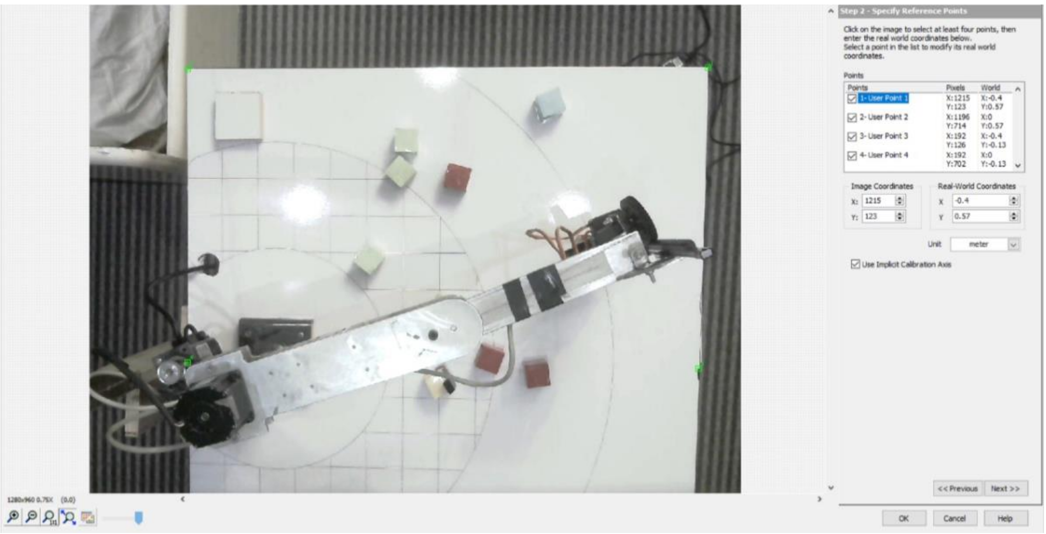


Figure 5.6: vision assistant block diagram.

Vision system’s strategy is applied using NI vision assistant program, the first step is to acquire an image from the camera, then the next step is to adjust the brightness’s value; to correct the colors value.

Image calibration is used to map the pixels in the image plane to the real-world units which in this case is in meters (see Figure 5.7), then the coordinate system is set according to the real world’s value of the points (see Figure 5.8).

After setting the coordinates for the base frame, objects detection is done by color pattern matching block, this block uses a stored picture to compare it with the possible objects within the region of interest (see Figure 5.9), the region of interest is within the green rectangle. The found matches are marked in red squares, the objects poses are found with respect to the base frame of the robot.



Step 2 - Specify Reference Points

Click on the image to select at least four points, then enter the real world coordinates below.
Select a point in the list to modify its real world coordinates.

Points

Points	Pixels	World
☑ 1- User Point 1	X:1215 Y:123	X:-0.4 Y:0.57
☑ 2- User Point 2	X:1196 Y:714	X:0 Y:0.57
☑ 3- User Point 3	X:192 Y:126	X:-0.4 Y:-0.13
☑ 4- User Point 4	X:192 Y:702	X:0 Y:-0.13

Image Coordinates: X: 1215, Y: 123
Real-World Coordinates: X -0.4, Y 0.57
Unit meter
☑ Use Implicit Calibration Axis

Figure 5.7: Image calibration reference points.

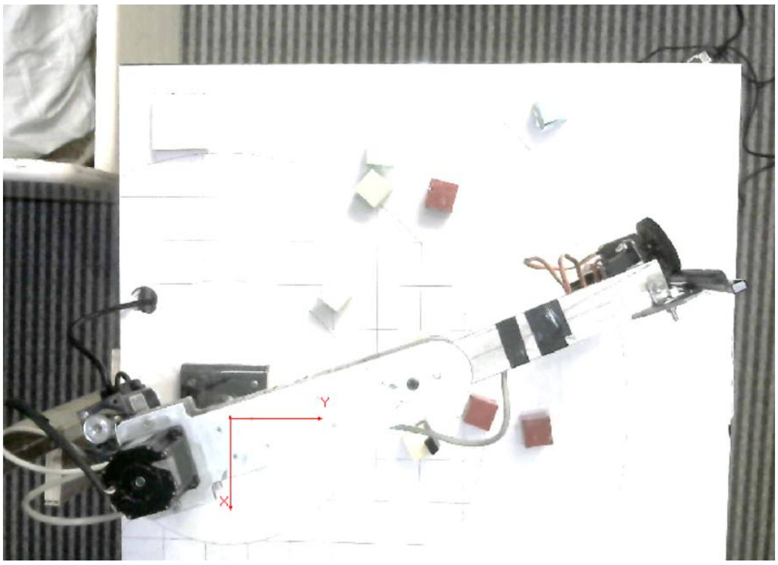


Figure 5.8: base frame's coordinate system after image calibration.

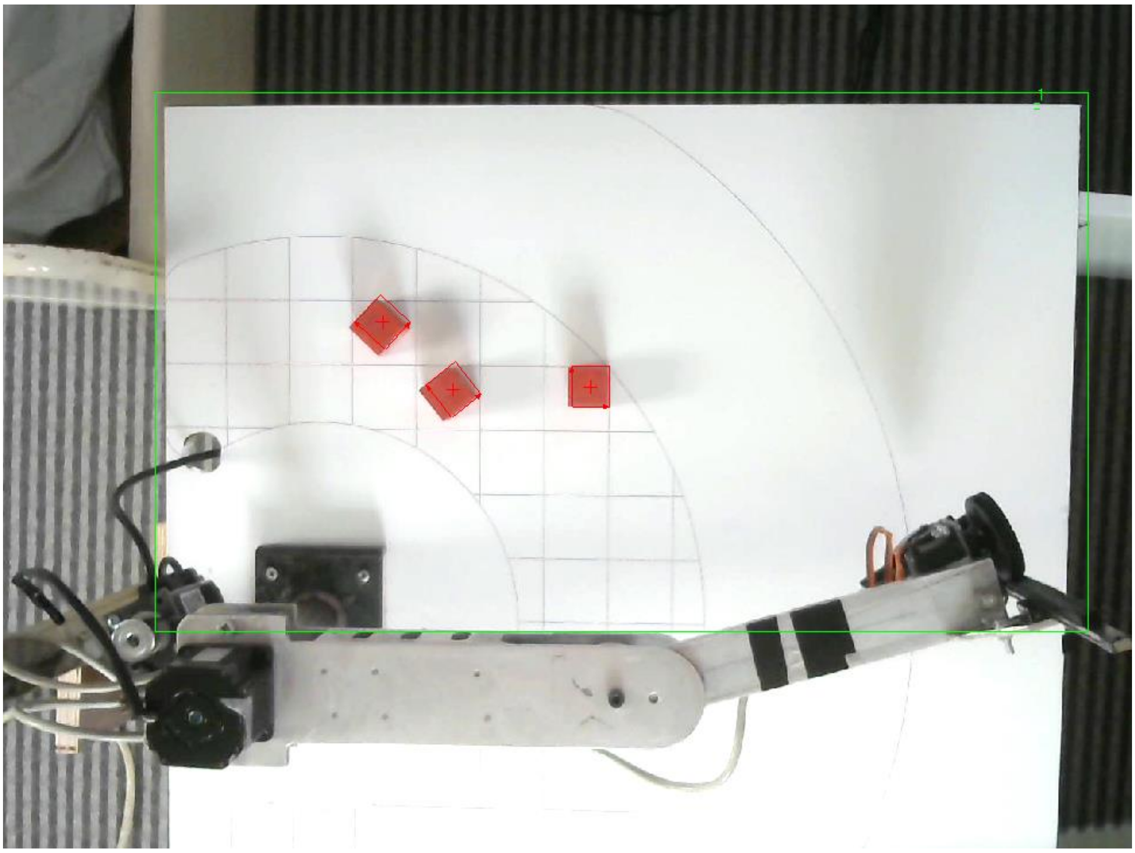
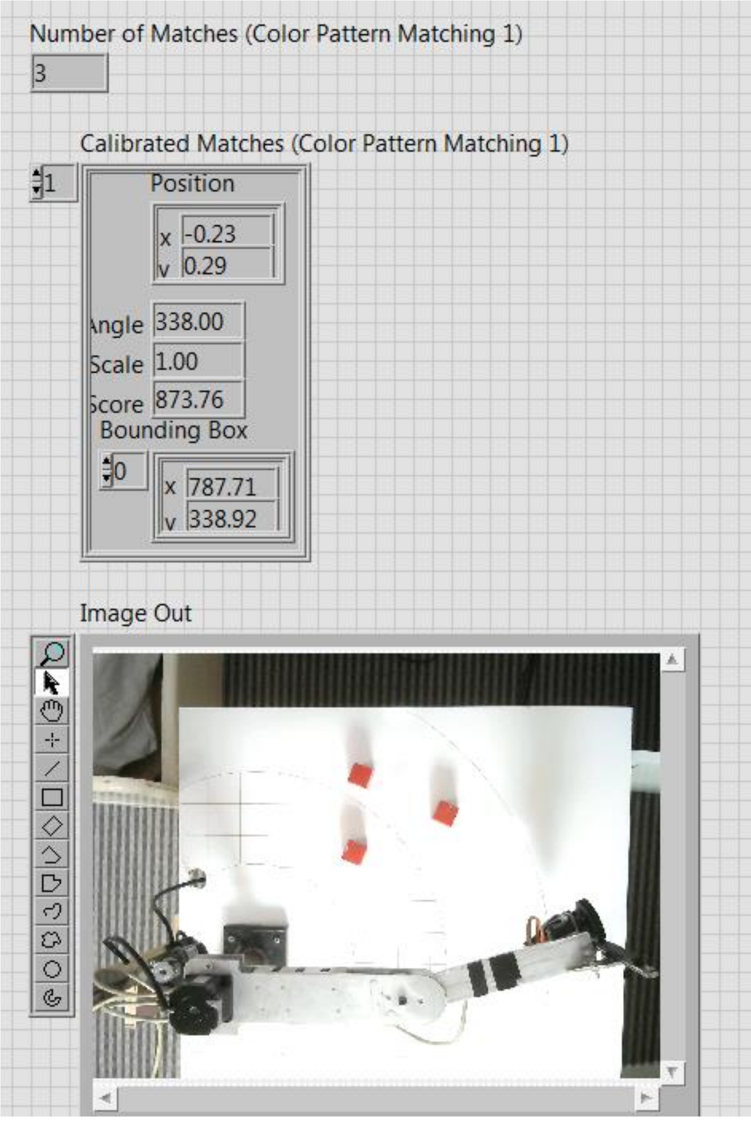


Figure 5.9: Color pattern matching

The previous processes are applied on images only, in order to apply it on a video, it must be connected to a video acquisition program in LabView's environment, which in result can be applied as a real-time software (see Figure 5.10).



Number of Matches (Color Pattern Matching 1)
3

Calibrated Matches (Color Pattern Matching 1)
1

Position
x -0.23
y 0.29

Angle 338.00
Scale 1.00
Score 873.76

Bounding Box
0
x 787.71
y 338.92

Image Out

Figure 5.10: Front panel for real-time application of objects detection.

Figure 5.10 shows the front panel for the application, that designed to provide each detected object pose for a specific color. The strategy that used here is to sort objects with the same color individually regardless to other objects with other colors.

Integration of the vision system with the invers kinematic algorithm to actuate the robot autonomously is not done yet.

6

Chapter 6

Control Architecture and Motion Control simulations

6.1 Control Architecture

The control system to supervise the activities of a robotics system must provide manipulation ability in the working environment, sensory ability, data processing ability and may have intelligence ability [8].

The general industrial robotics system's control architecture is considered as superposition of several activity levels arranged in hierarchical structure which detailed in Figure 6.1 [8].

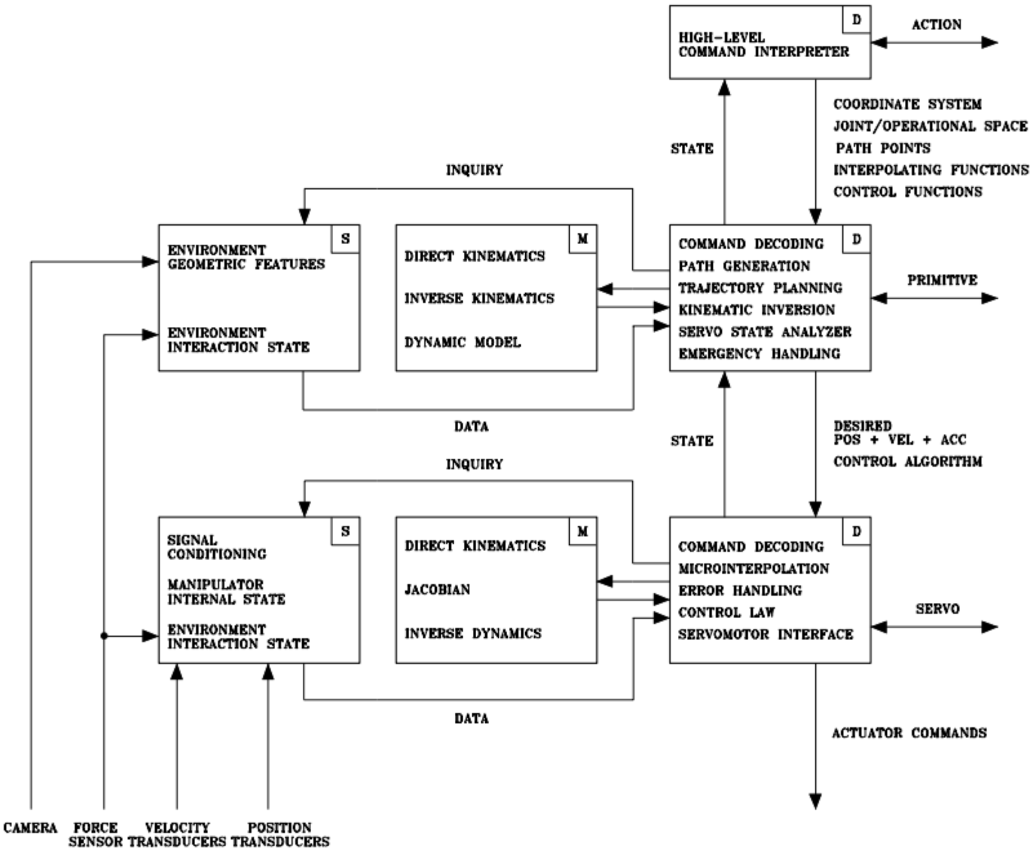


Figure 6.1: Hierarchical functional architecture for industrial robotics system [8].

At action level, with referring to the knowledge base of the robot dynamics and links mechanical limits also with referring to data yielded from the vision system, decisions will be made about the operation of links movement. These decisions will decide whether the motion plane will be in joint space or in operational space, also decide the strategy of collision avoidance [8]. In this project the vision system will provides the programmed LabView robotics application each object's color and its pose in the robot operational space. An inverse kinematics algorithm used to compute the joint variable to achieve a desired end-effector pose, with taking into the account the robot mechanical limits.

At primitive level, the decisions made in the action level will be computed and detailed in this level where the decided trajectory will be computed with time, to be as a reference for the control algorithm in the *servo level.* Also, it defines the control algorithm [8].

At servo level, the control law must guarantee the actuators to track the specified trajectories in the *primitive level*, where the control algorithm operates on the error signal between the controller reference and the actual measured positions or velocities [8].

The actions flowing in this project on the basis of the in Figure 6.2 flow chart:

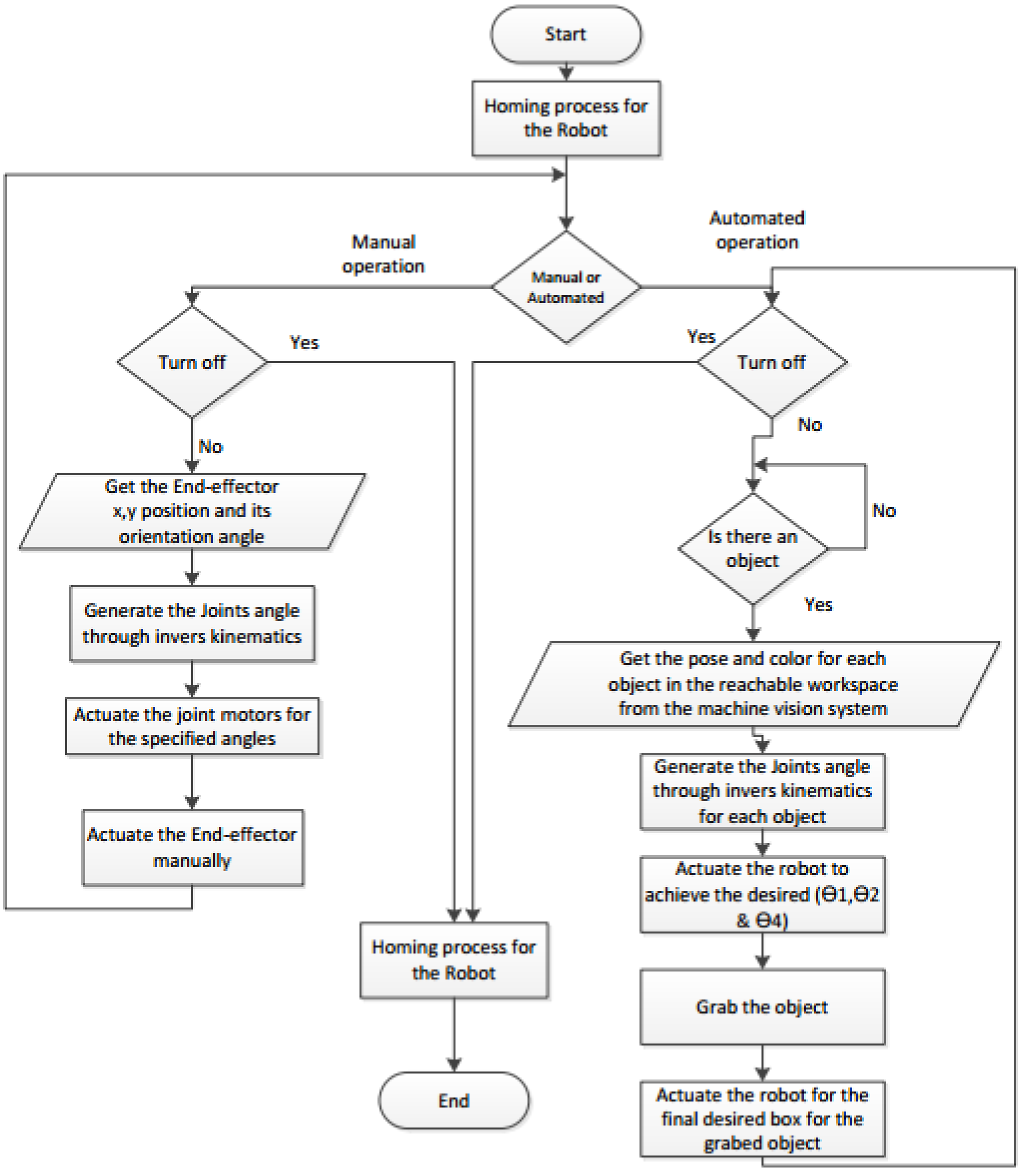

Start
Homing process for the Robot
Manual operation
Automated operation
Manual or Automated
Turn off
Yes
No
Get the End-effector x,y position and its orientation angle
Generate the Joints angle through invers kinematics
Actuate the joint motors for the specified angles
Actuate the End-effector manually
Homing process for the Robot
End
Yes
Turn off
No
No
Is there an object
Yes
Get the pose and color for each object in the reachable workspace from the machine vision system
Generate the Joints angle through invers kinematics for each object
Actuate the robot to achieve the desired (Ө1,Ө2 & Ө4)
Grab the object
Actuate the robot for the final desired box for the grabed object

Figure 6.2: Working flowchart.

6.2 Motion control

Robot manipulator's controllers decide the joint torques/forces that are needed to make the mechanical structure track a desired motion plan (trajectory with time) with a desired dynamic performance. The design process of these controllers will be based on the dynamic model of the manipulator [8]. The control scheme will be in the joint space, where the desired end-effector pose will be mapped to the joint variables, theses joint variables will act as the reference for the controller [8].

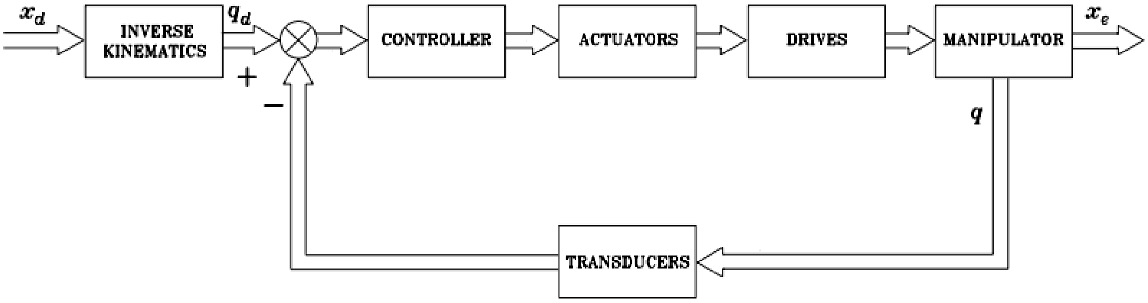


Figure 6.3: General scheme of joint space control [8].

The disadvantage of joint space control that the end effector pose will be controlled as an open loop, where there are no direct measurements for end effector pose, in other words it will be determined through direct kinematics depends on the joints measurements. Hence the end effector pose estimation will be affected due to inaccuracy in joint measurements [8].

In this project, two control strategies will be used in the joint space, decentralized control (independent joint control) and centralized control (dependent joint control).

6.2.1 Decentralized control

In this control strategy each link will be controlled independently to the others. The nonlinear configuration-dependent forces (centrifugal and coriolis) will be considered as disturbances act on the plant. The system is highly nonlinear and cannot be linearized due to high range of the operating points, but in this case the independent joint controllers will be linear controllers with good disturbance rejection characteristics.

The mass matrix will be decoupled into constant terms and configuration-dependent terms, this decoupling shown in Eq (51).

$$M(q) = \bar{M} + \Delta M(q) \tag{51}$$

$$\bar{M} = \begin{bmatrix} 0.1034 & 0 & 0 & 0 \\ 0 & 0.036 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1.32 * 10^{-5} \end{bmatrix} Kg.m^2, \tag{52}$$

where $\bar{M}$ is the diagonal matrix whose elements are constant average inertia at each joint [8]. Each one of these elements will be modeled as mechanical system. We assume that each one of this mechanical system need to be controlled to have a desired dynamic performance and minimum steady state error. This yield to design four linear controllers for this SCARA robot.

The linear controller considered as PD-feedback compensator, because of its good disturbance rejection characteristic.

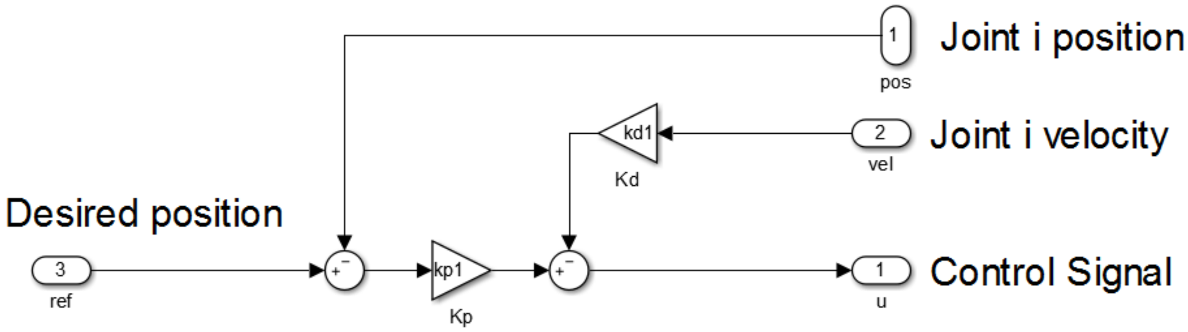


Figure 6.4: PD controller (feedback compensation) for each joint.

Where Eq (53) represents the transfer function corresponding each constant average inertia element of matrix $\overline{M}$, Eq (54) represents the closed-loop transfer function. The desired dynamic performances for each link supposed to have maximum overshoot of 10% with maximum settling time 0.1 $(second)$.

$$G(s) = \frac{1}{\overline{M}_{ii}s^2}, \tag{53}$$

$$T(s) = \frac{K}{\overline{M}_{ii}s^2 + K_f s + K}, \tag{54}$$

$$K_1 = 1\,,$$

The general second order transfer function is given by [18]:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{55}$$

$$T_s = \frac{4}{\xi\omega_n} \tag{56}$$

$$\%OS = e^{-(\xi\pi/\sqrt{1-\xi^2})} \times 100 \tag{57}$$

From the desired dynamic performance, ω_n and ξ will be found from Eq (56) and Eq (57), where $\xi = 0.59$ and $\omega_n = 67.6\,rad/second$. The desired second order characteristics equation for each link must equal Eq (58).

$$\Delta = s^2 + 5391.4s + 4569.7, \tag{58}$$
$$\Delta_i = \bar{M}_{ii}s^2 + K_{fi}s + K_i, \tag{59}$$

The gains of the controllers can be found by matching each link characteristic equation (Eq (59)) with the desired second order characteristics equation (Eq (58)),

Table 6.1: The gains for each controller.

	$\boldsymbol{K}$	$\boldsymbol{K_f}$
Controller 1 ($\bar{M}_{11}$)	472.4	8.27
Controller 2 ($\bar{M}_{22}$)	164.4	2.88
Controller 3 ($\bar{M}_{33}$)	2284	40
Controller 4 ($\bar{M}_{44}$)	0.0603	0.0011

6.2.2 Computer simulation and results

The SCARA robot is simulated using MATLAB/Simulink (see Figure 6.5 and Figure 6.6), where the highly nonlinear dynamic model of the manipulator is represented in Simulink using S-Function (see Appendix C) without linearization.

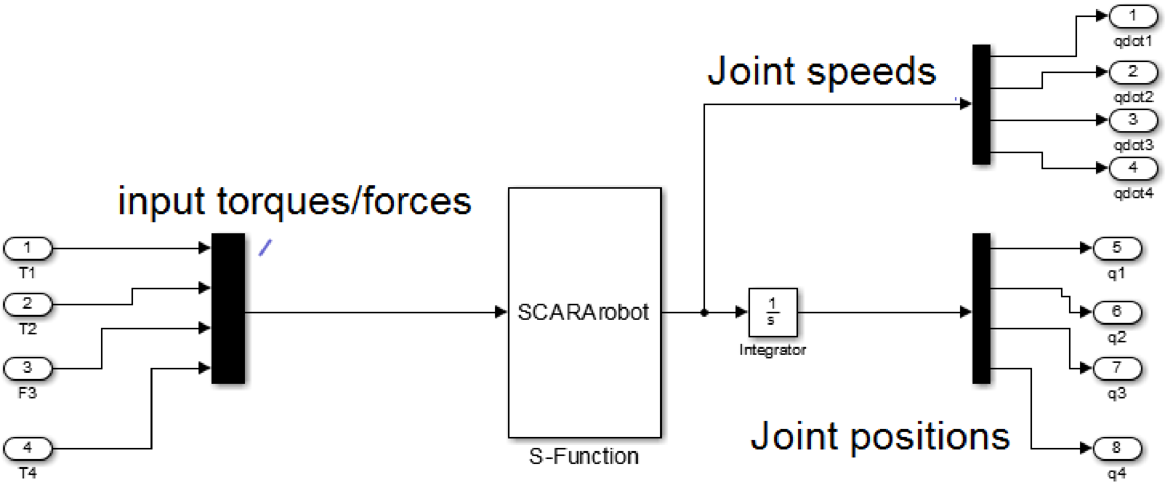


Figure 6.5: S-function Simulink model for the SCARA robot.

The trajectories which will be the reference point for the controllers in the simulation are $\vartheta_1(0° \: to \: 120°)$,$\vartheta_2(0° \: to \: 30°)$,$d_3(0cm \: to \: 18cm)$, and $\vartheta_4(0° \: to \: 57°)$ in 1.5 second. A trapezoidal velocity profile for the trajectory is planed here in order not to face a sudden acceleration or deceleration in the trajectory. Figures 5.7, 5.8 and 5.9, Figure 5.10 show how the controllers track the desired trajectory with accepted accuracy.

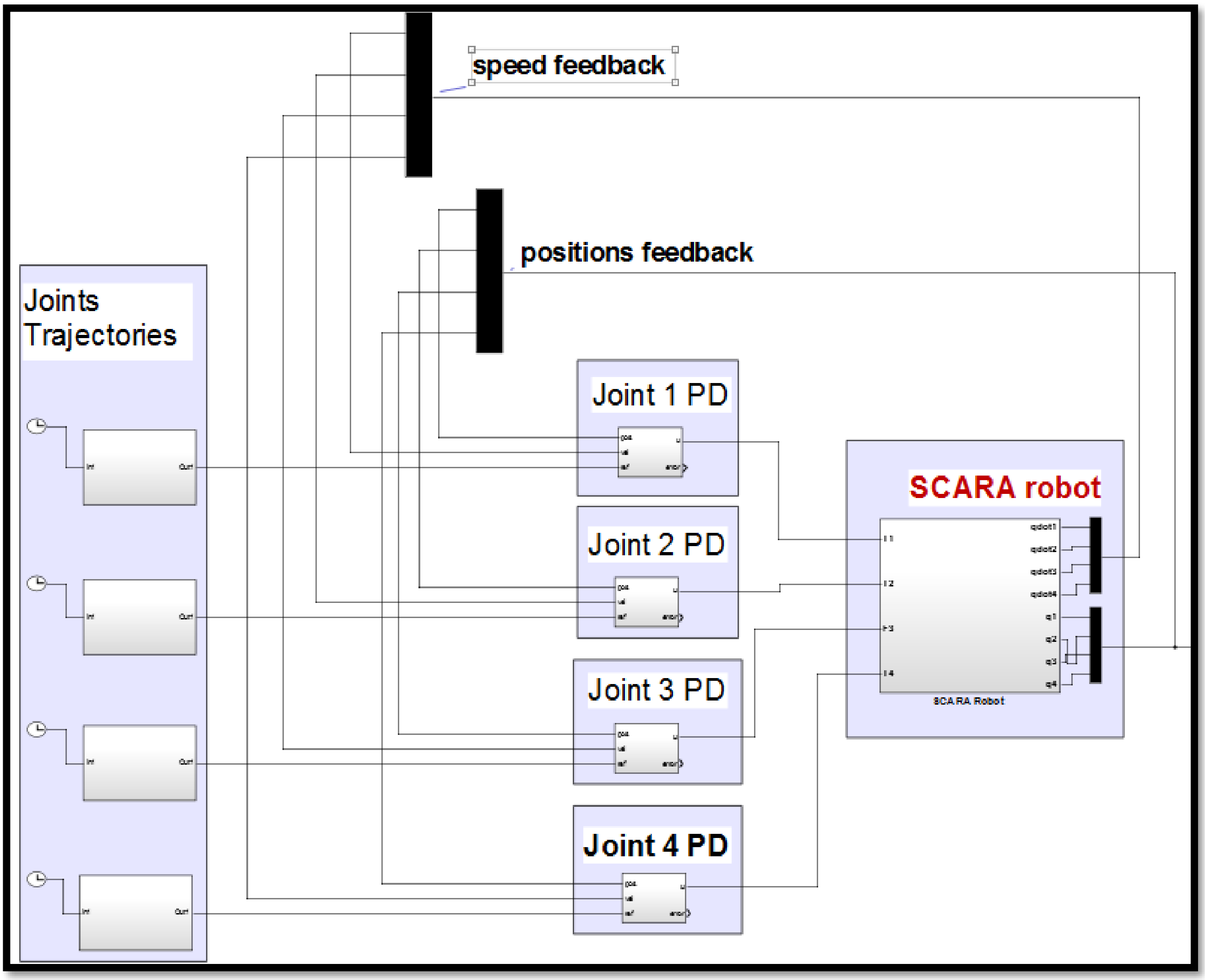


Figure 6.6: Simulink model for the robot with its controllers**.**

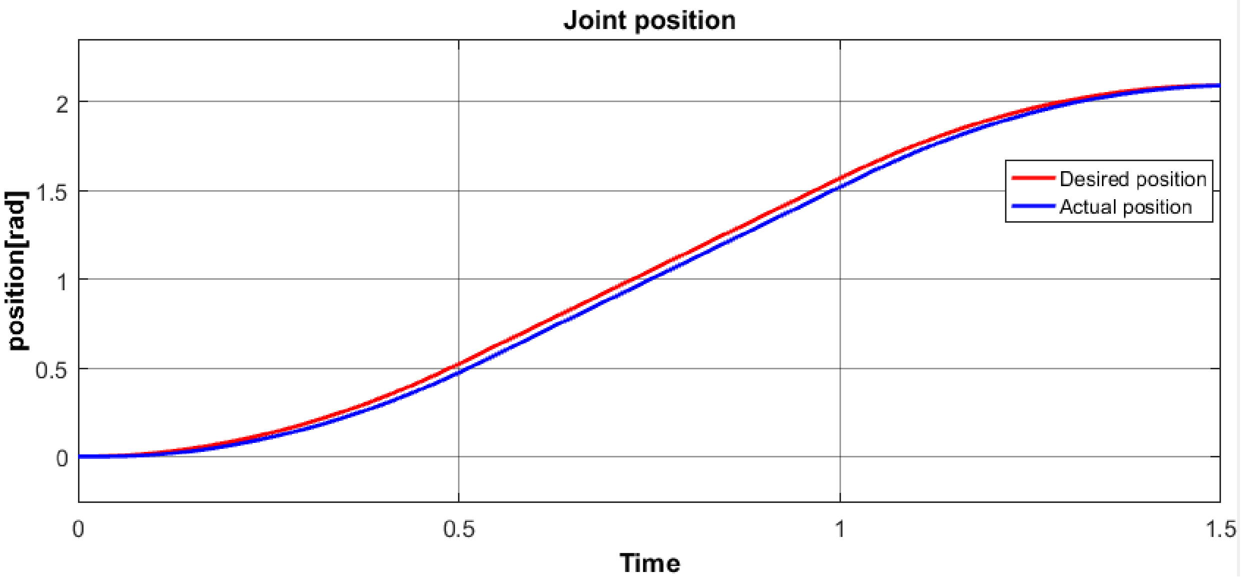


Figure 6.7: Joint 1 actual and desired positions.

The maximum error here is occurred at the midpoint of time duration with a magnitude of $0.0336\ rad$, eventually the steady state error goes to 0.

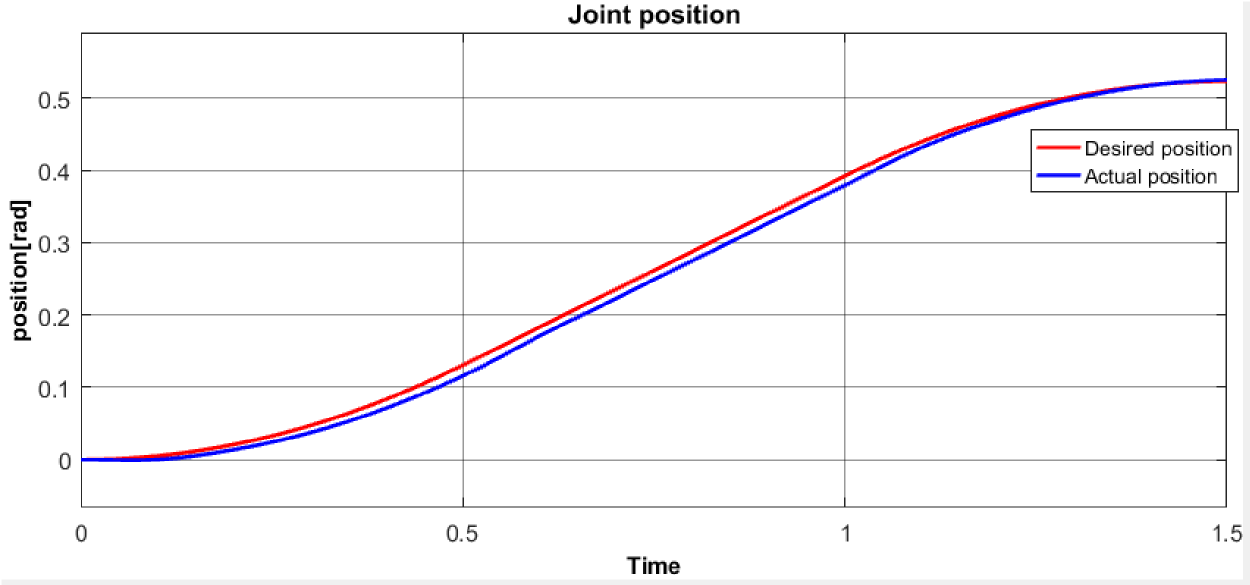


Figure 6.8: Joint 2 actual and desired positions.

The maximum error here is occurred at the $0.5\ s$ with a magnitude of $0.01117\ rad$, eventually the steady state error goes to$-1.6\ x\ 10^{-3} rad$.

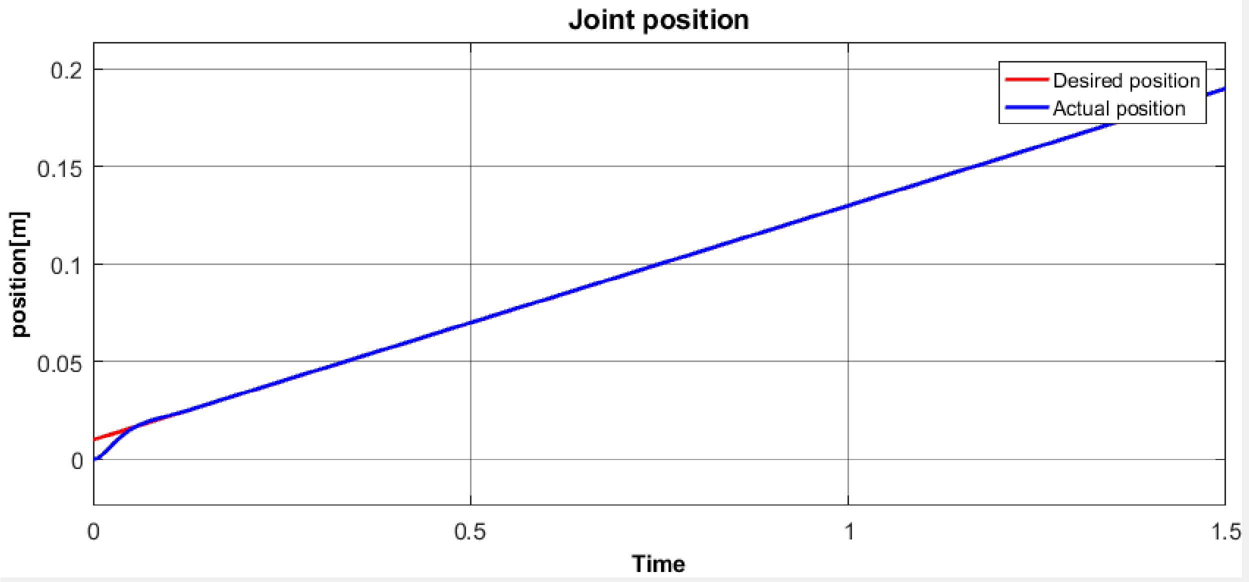


Figure 6.9: Joint 3 actual and desired positions.

Here, the steady state error goes to $-4.4\ x\ 10^{-5}\ m$.

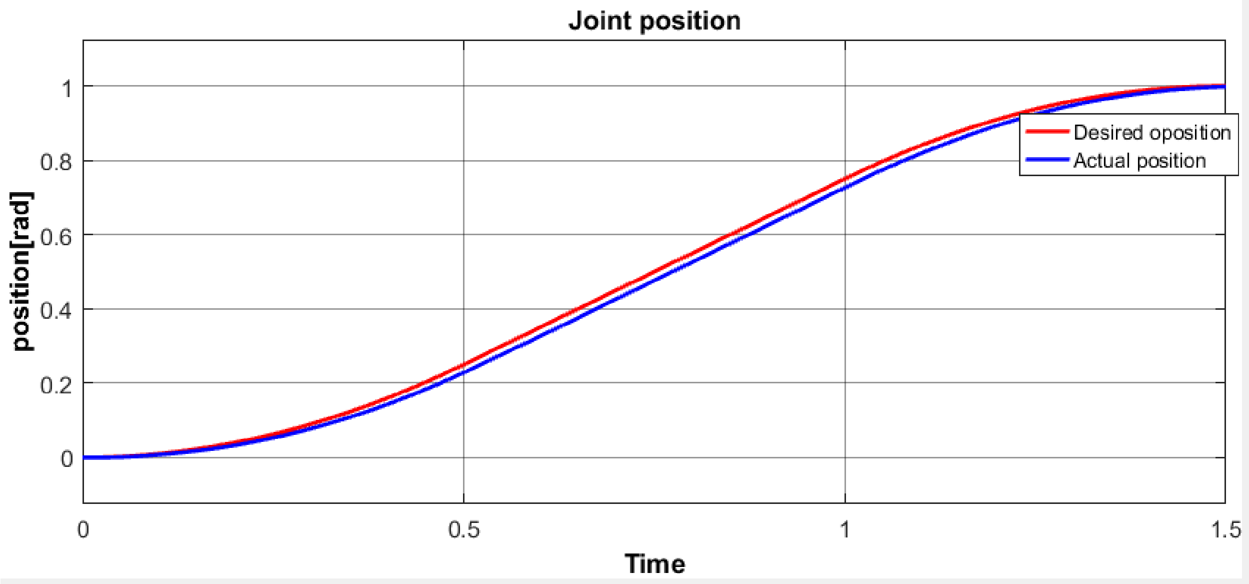


Figure 6.10: Joint 4 actual and desired positions.

The maximum error here occurred at the $0.5\ second$ with a magnitude of $0.01117\ rad,$ eventually the steady state error goes to$-1.6 \times 10^{-3} rad$.

It is notable how the error occurred due to the nonlinearity of the system, also this simulation is based on dynamical model which is derived with the assumptions of neglecting the friction between links as well as the links are completely rigid. Therefore, the real-world implementation for these controllers to control such manipulator will arise greater amount of error than the simulation error.

Figures 6.10-6.14 show the control signal corresponding each joint, where the results show a small deviation from the yielded results in the (motor sizing section).

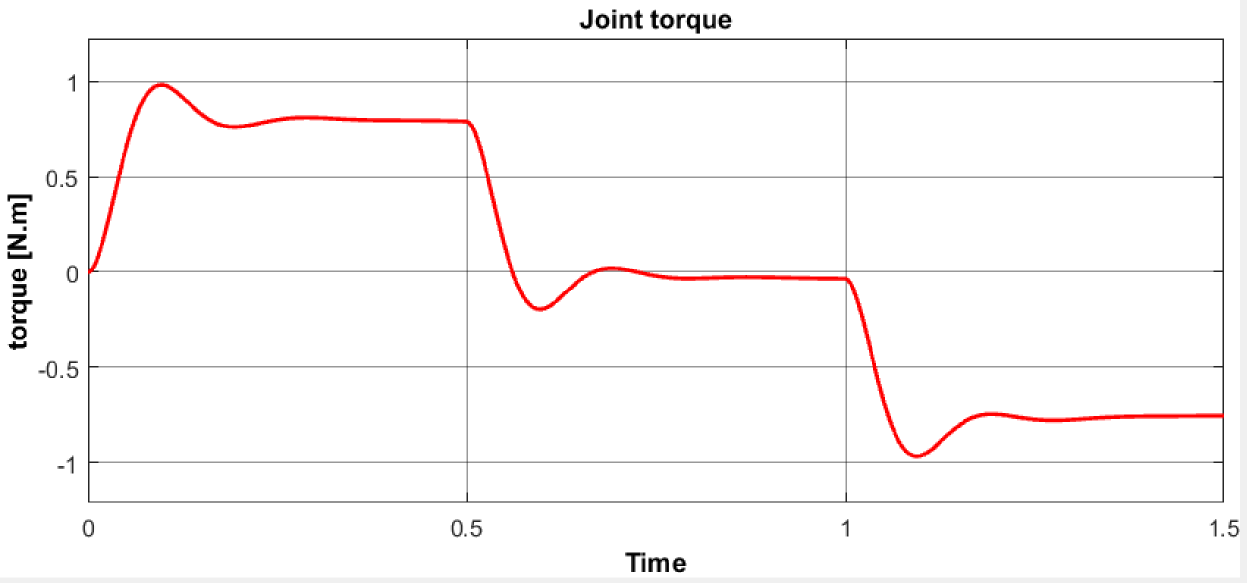


Figure 6.11: Joint 1 control signal (Motor1 input torque).

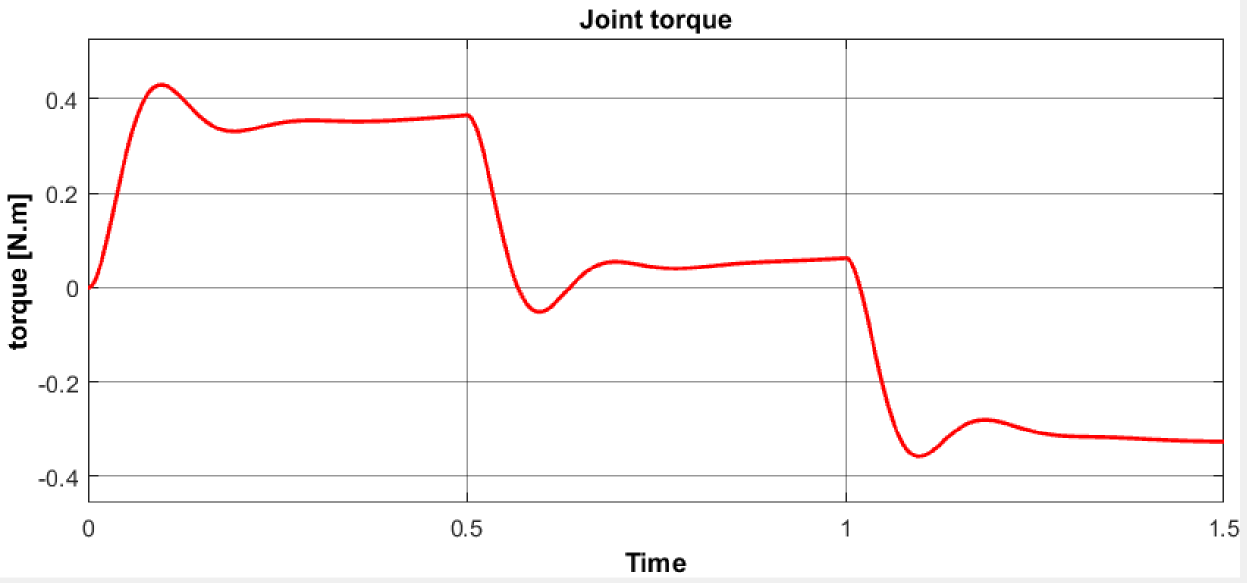


Figure 6.12: Joint 2 control signal (Motor2 input torque).

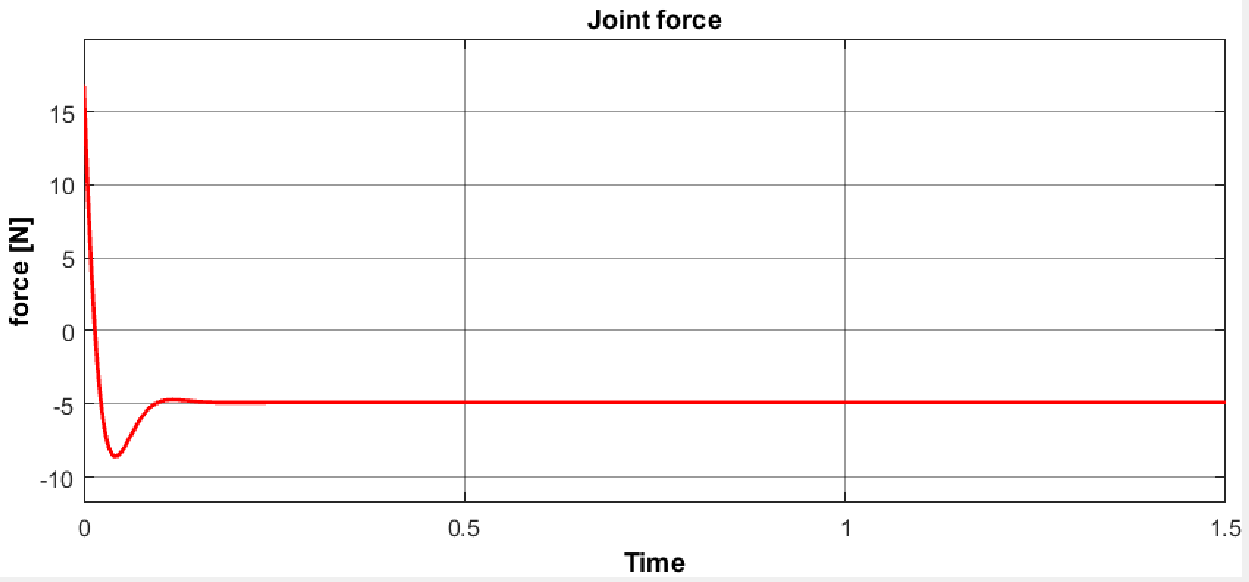


Figure 6.13: Joint 3 control signal (Motor 3 input force).

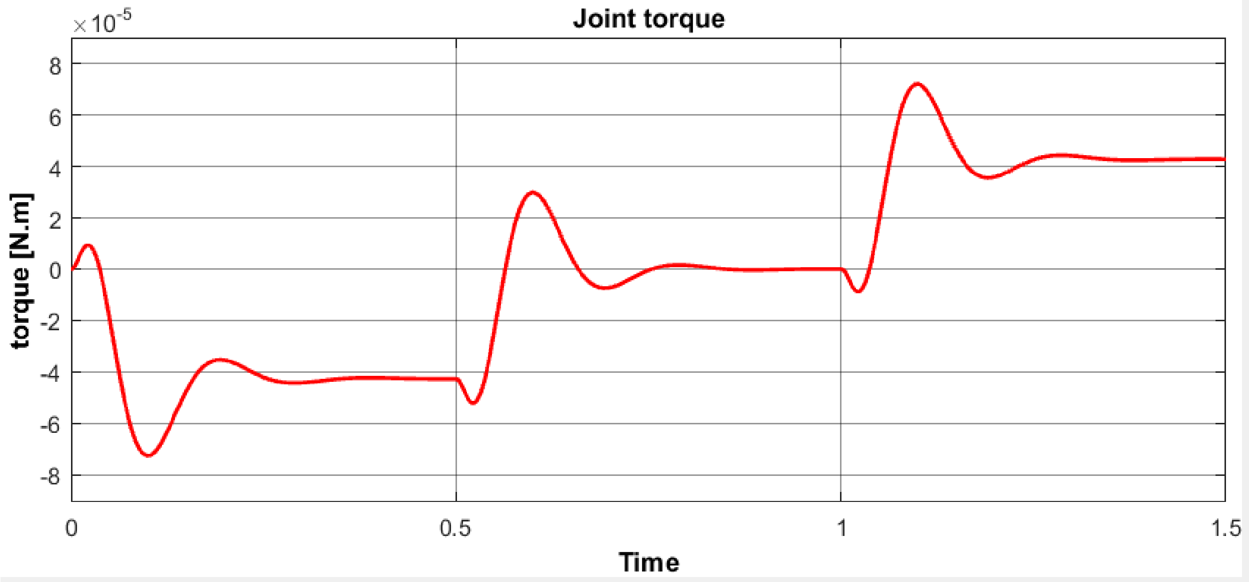


Figure 6.14: Joint 4 control signal (Motor 4 input torque).

Figures 6.15-6.18 show the desired trapezoidal velocity profile and the actual profile for each joint.

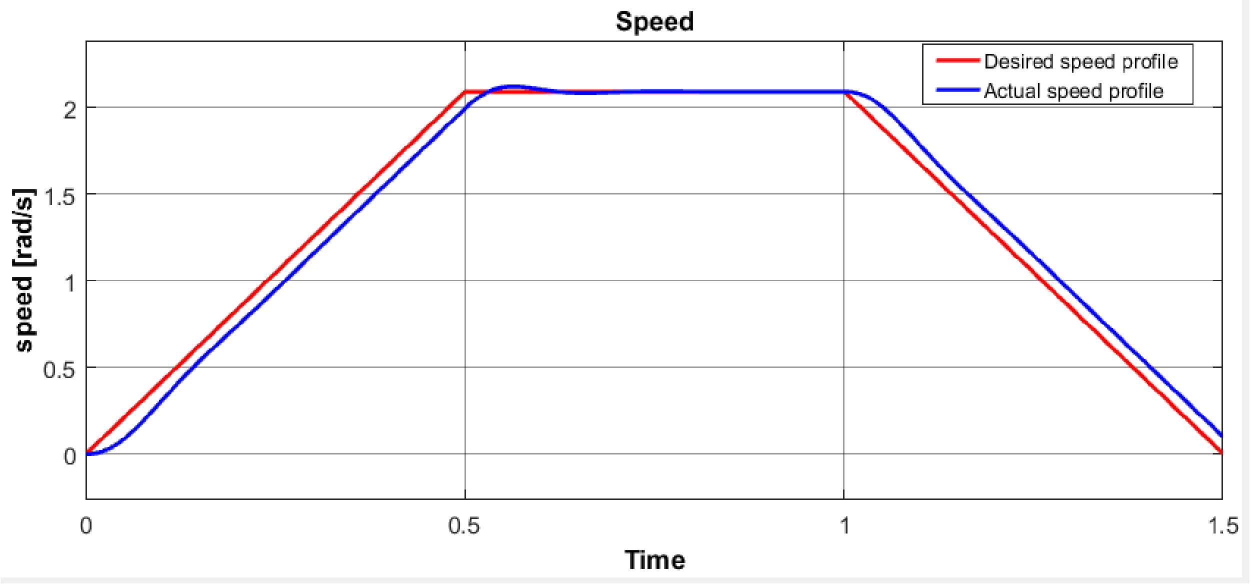


Figure 6.15: Joint 1 actual and desired velocity.

Here, the steady state error in the ramp stage equal $0.068\ rad/second$.

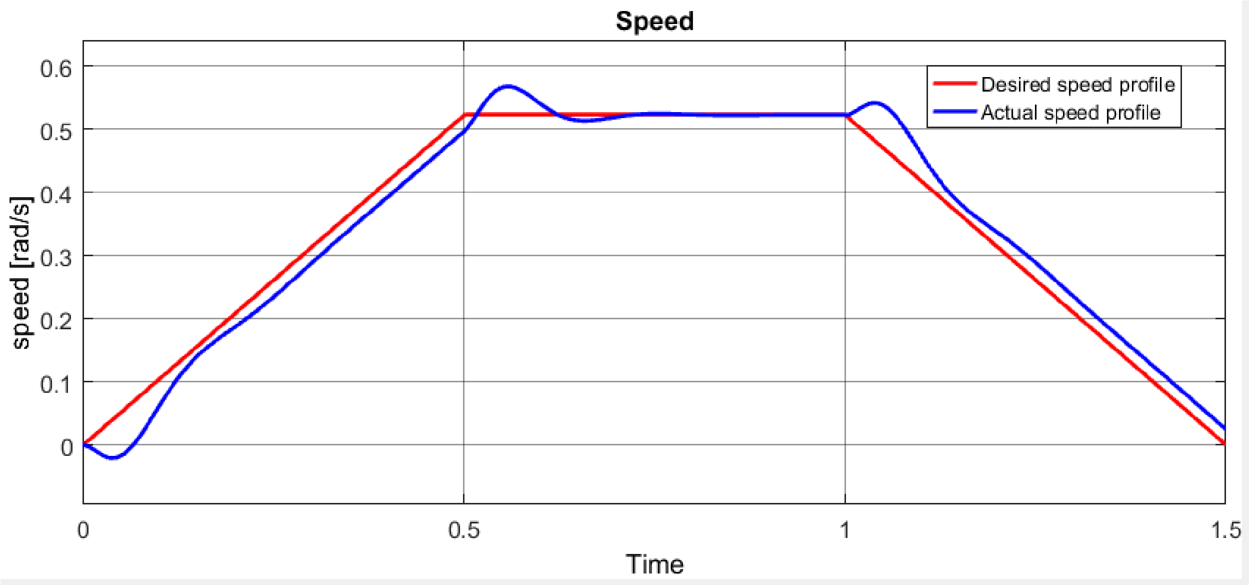


Figure 6.16: Joint 2 actual and desired velocity.

Here, the steady state error in the ramp stage equal $0.0196\ rad/second$.

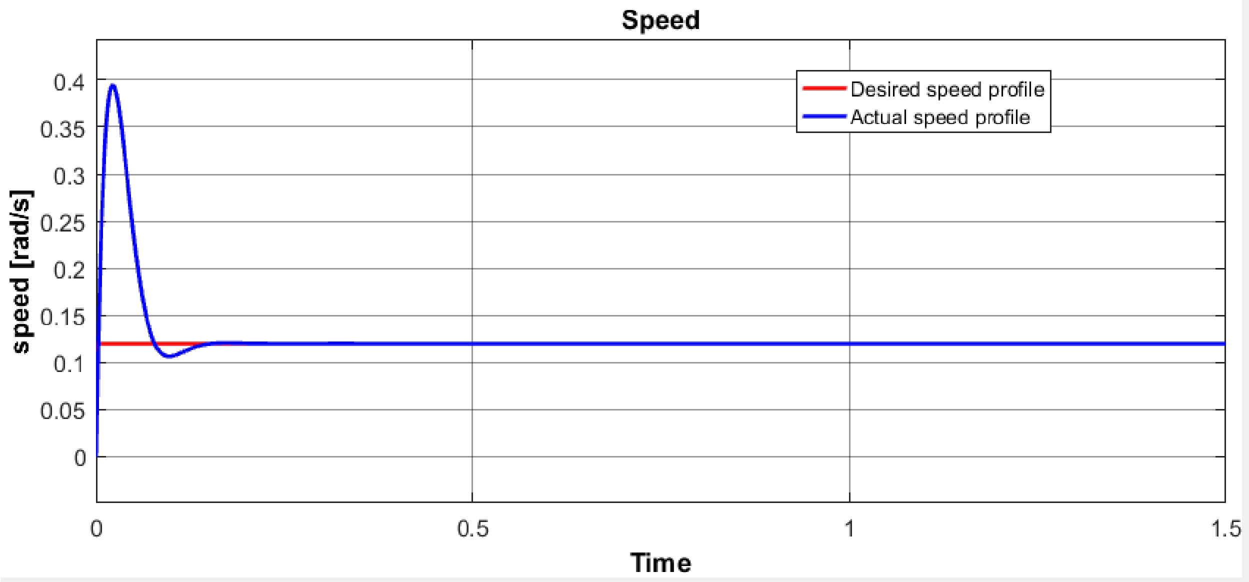


Figure 6.17: Joint 3 actual and desired velocity.

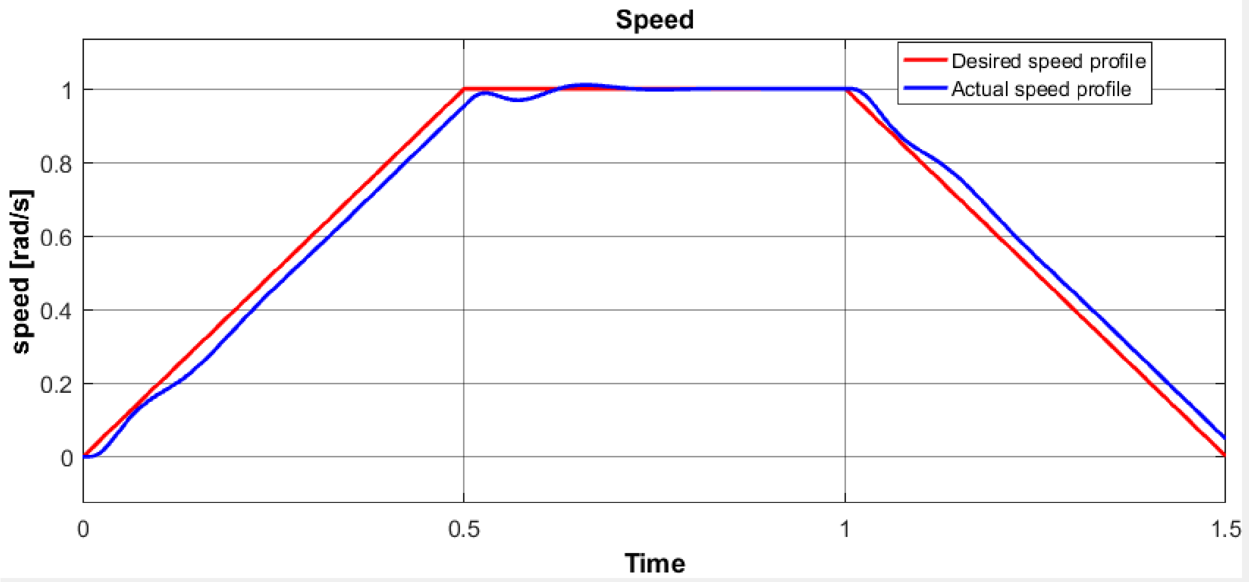


Figure 6.18: Joint 4 actual and desired velocity.

Here, the steady state error in the ramp stage equal $0.034\ rad/second$.

It is advisable to introduce a centralized nonlinear control algorithm to compensate the nonlinear effect of the plant like nonlinear-feedback control. Here the dynamic performance expected to be better than the decentralized, where the controller is expected to provide better robustness and better error compensation.

Simulation discussion

The introduced independent controllers show a good dynamic performance, good tracking ability and good disturbance rejection for such a nonlinear system, due to the simple structure of the SCARA robot, where there are no gravitational loads in Joint 1, Joint 2 and Joint 4.

Here the maximum torque arises for Joint 1 $\approx 1N.m$ and $\approx 0.41\ N.m$ for Joint 2, where these results are below the assumed torques in (motor sizing section).

7

Chapter 7

Implementation of the project

7.1 Software application

The robotics system is programmed under LabView software, where the algorithm detailed in Figure (6.2) is translated as LabView code. A state machine structure will contain three main states to operate the project, these states are homing state, manual operation state and automated operation state, the transitions between the states allow the functionality of flowing the data as detailed in Figure (6.2).

The homing operation procedure is to rise the end-effecter upward, open the gripper then rotate the wrist joint to its zero-angle position. The next step is to rotate the First joint to the left until it reaches the left limit switch to ensure that it works properly for safe manipulation, then turn the First link to the right until it reach the right limit switch, here the first link is ready to go to its zero-angle position. After homing the first link, the same procedure is used for the Second link; the Homing operation is mandatory when turning the system on or off to ensure that the robot start to operate from the zero position. The locations of the limit switches for the First link and Second link are shown in Figure 7.1 and Figure 7.2, respectively.

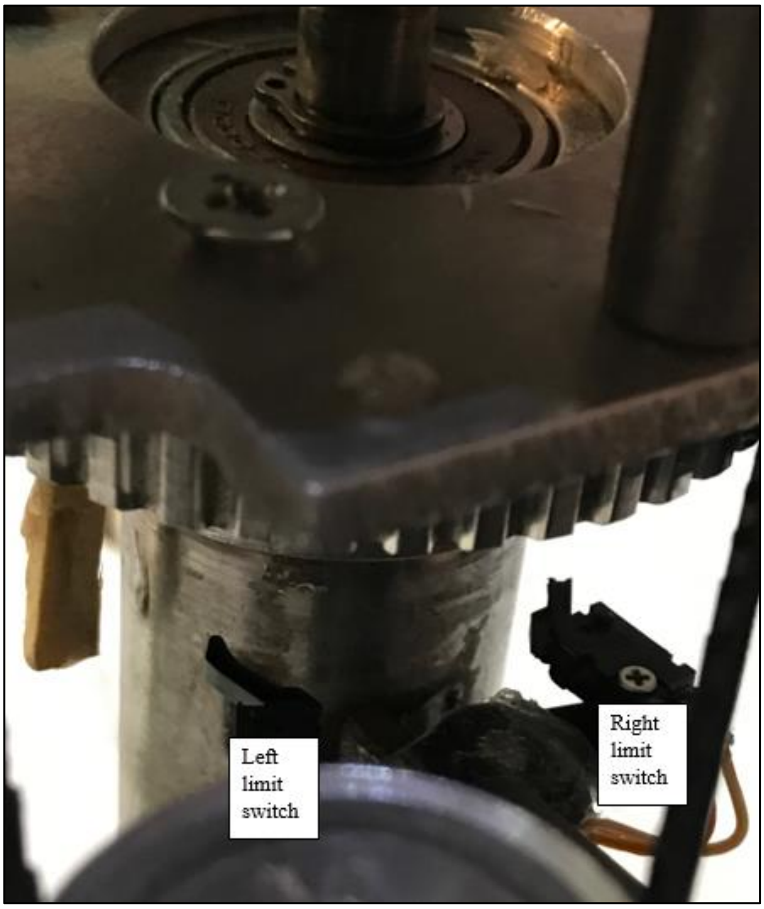


Figure 7.1: First link limit switches.

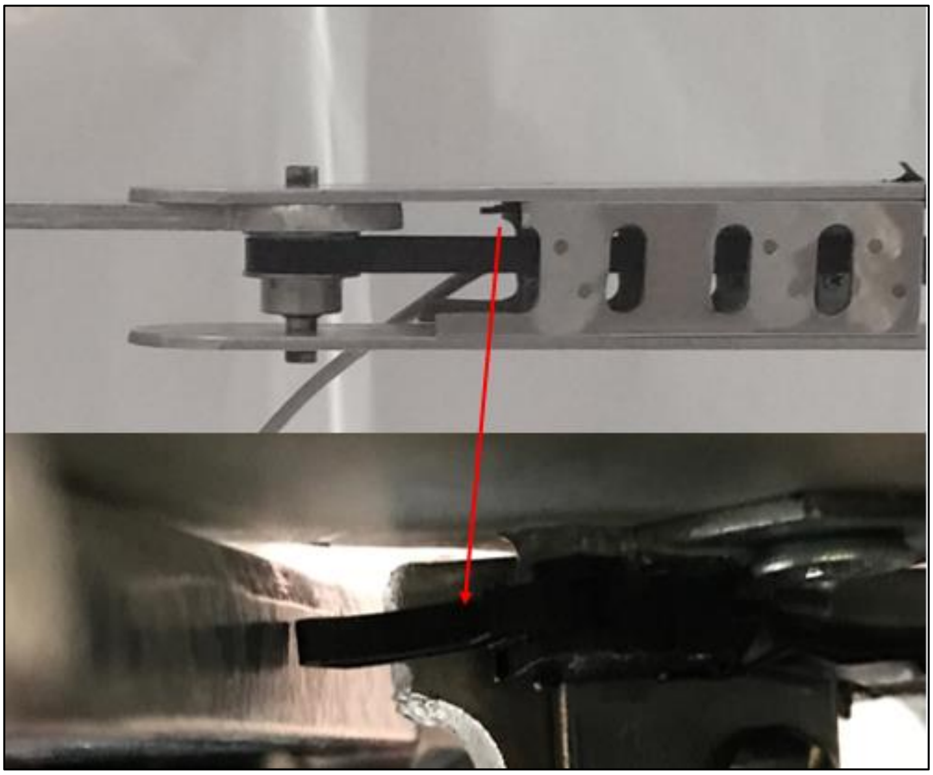


Figure 7.2: Second link left limit switch.

The manual operation is to actuate the robot as the user demands. The user will be asked for an achievable end-effector pose, the actuator will be actuated to achieve the demanded pose, where the inverse kinematics algorithm will decide the joints angles for the demanded pose. The Third prismatic joint and the gripping mechanism will be actuated manually.

7.2 The structure of the robotics system

Figure 7.3 shows a whole view for the project, Figure 7.4 shows a top view for the robot with its working space, also it illustrates the zone of the random objects and zones for sorted objects.

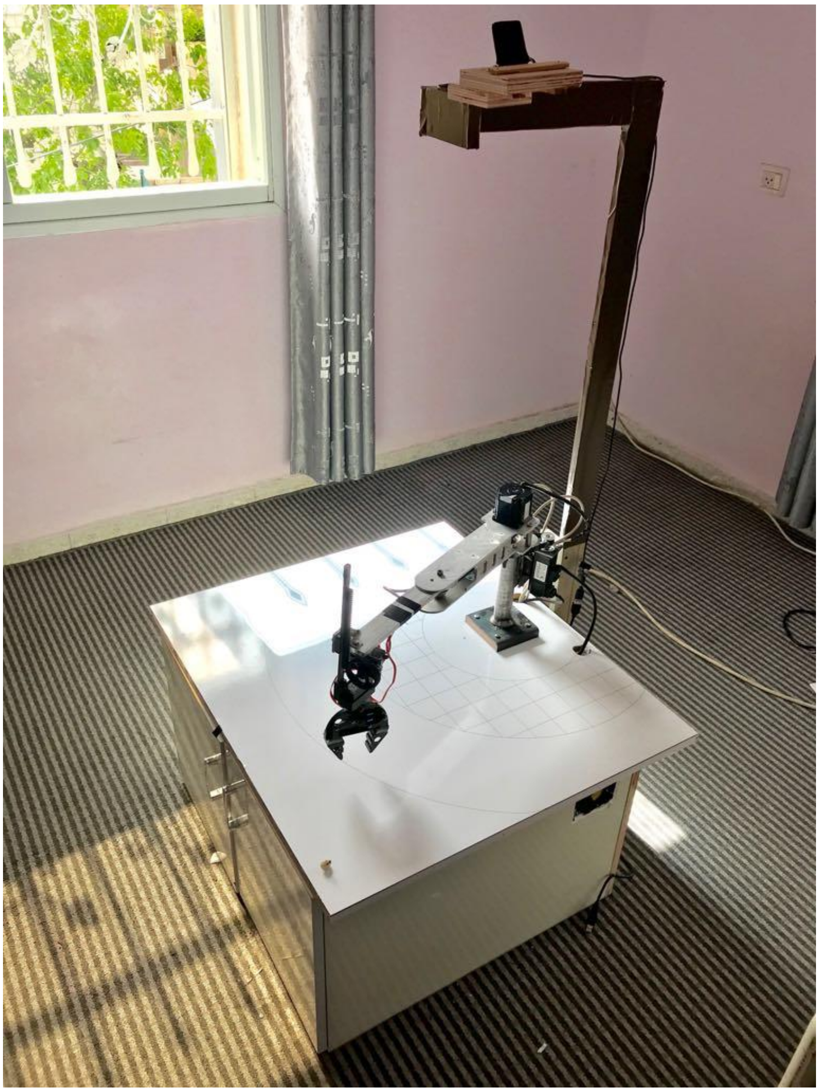


Figure 7.3: The whole structure of the robotics system.

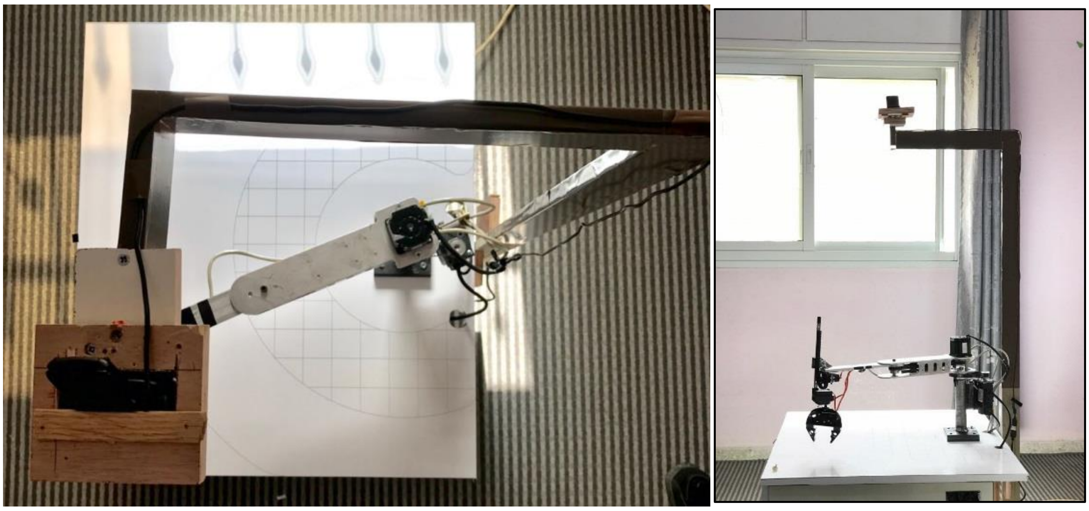


Figure 7.4*:* Top and Side view of the camera**.**

7.3 Recommendations and results

Recommendations

This section will provide recommendations to improve the implanted robotics system in this project, these recommendations will be presented as follow: recommendations for system components, recommendations for the vision system, recommendations for the mechanical structure of the robot and recommendations for the control algorithm of the robot.

Recommendations for system components:

The motor of the Second link is better to be replaced to a servo-motor instead of stepper-motor to yield a better performance and to reduce the noise, also to get a more accurate positioning. Another suggestion is to use hybrid-steppermotors for the First and Second links, where it is cheaper than using servo-motors and it could be accurate as servo-motor because the strategy of driving of a hybrid-steppermotor is micro-stepping, also the angle of the motor shaft is feedbacked through a rotary encoder.

The Third prismatic joint actuator and the Wrist joint actuator must not be actuated using the MG996R servomotors because they are very sensitive to the noise, and they are not designed to work in industrial environment. DC-motors with rotary encoders or potentiometer could be used to actuate the Third and Wrist joints.

Recommendations for the vision system:

The vision process is implemented to work offline with the actuation loop of the robot, it is preferable to let the vision system to provide the data in online to actuate the robot.

Recommendations for the mechanical structure of the robot:

The mechanical joints and the mechanical structure of the robot show smooth movements; however, the structure could be improved to yield a better performance.

Recommendations for the control algorithm of the robot:

If the Second link actuator replaced with a servomotor, the robot joint angles feedback will be available to implement any designed control algorithm like decentralized PID control algorithm, centralized invers dynamic control, centralized robust control or any robot control algorithm.

Also, the robot can be controlled in the operational space, where the measurements in the operational space can be extracted from the vision system.

Results:

LabView application is programed to control the robotics system, where this application designed in the state-machine and sequence of action fashions. The homing process worked properly. The manual operation is programmed to actuate the robot manually, where the user is asked to enter the end-effector pose, then he or she can actuate third joint and the gripping manually.

The automated process worked, but not in its full-mode. The implementation of the automated process was through a sequence of actions repeated in a loop, where the first action in the sequence is to get the pose of the object, then this information will be used in the next action to generate the joint variables through the inverse kinematic algorithm to actuate the robot arm. The robot is actuated through the data from the vision system with bad accuracy $(\pm 2\ cm)$, so that the vision system must be enhanced to compensate this bad accuracy in position.

References

[1] L. Westerlund, *The extended arm of man: a history of industrial robot*. Informationsförlaget, 2000.

[2] (20-MARCH). *SOLIDWORKS*. Available: https://www.solidworks.com/

[3] (20-MARCH). *MATLAB*. Available: https://ww2.mathworks.cn/en/products/matlab.html

[4] G. J. Agin, *Real time control of a robot with a mobile camera*. SRI International, 1979.

[5] H. Makino and N. Furuya, "Selective compliance assembly robot arm," in *Proceedings of the First International Conference on Assembly Automation*, 1980, pp. 77-86.

[6] M. S. Alshamasin, F. Ionescu, and R. T. Al-Kasasbeh, "Modelling and simulation of a SCARA robot using solid dynamics and verification by MATLAB/Simulink," *International Journal of Modelling, Identification and Control*, vol. 15, no. 1, pp. 28-38, 2011.

[7] H. Goya, K. Matsusaka, M. Uemura, Y. Nishioka, and S. Kawamura, "Realization of high-energy efficient pick-and-place tasks of scara robots by resonance," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2730-2735: IEEE.

[8] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London: Springer, 2009.

[9] P. Corke, *Robotics, Vision and Control*. Berlin: Springer, 2011.

[10] F. Fahimi, *Autonomous robots: modeling, path planning, and control*. Springer Science & Business Media, 2008.

[11] S. Y. Nof, *Handbook of industrial robotics*. John Wiley & Sons, 1999.

[12] R. G. Budynas and J. K. Nisbett, *Shigley's mechanical engineering design*. McGraw-Hill New York, 2008.

[13] S. Centinkunt, "Mechatronics with Experiments," ed: New Delhi: Wiley, 2015.

[14] delta. (2009, 20-FEBRUARY). *ASDA-B series user manual*. Available: http://www.delta.com.tw/product/em/motion/motion_servo/download/manual/ASDA-B_M_EN_20090628.pdf

[15] N. Instruments. (25-FEBRUARY). *User guide and specifications NI-myRIO-1900*. Available: http://www.ni.com/pdf/manuals/376047c.pdf

[16] geometrixar. (18-OCTOBER). *VISION SYSTEM*. Available: http://www.geometrixar.com/vision-system.html

[17] T. M. Anandan. (2013, 18-OCTOBER). *Robotics+Vision at a Glance: The Dos, Don'ts and Applications*. Available: https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Robotics-Vision-at-a-Glance-The-Dos-Don-ts-and-Applications/content_id/4189

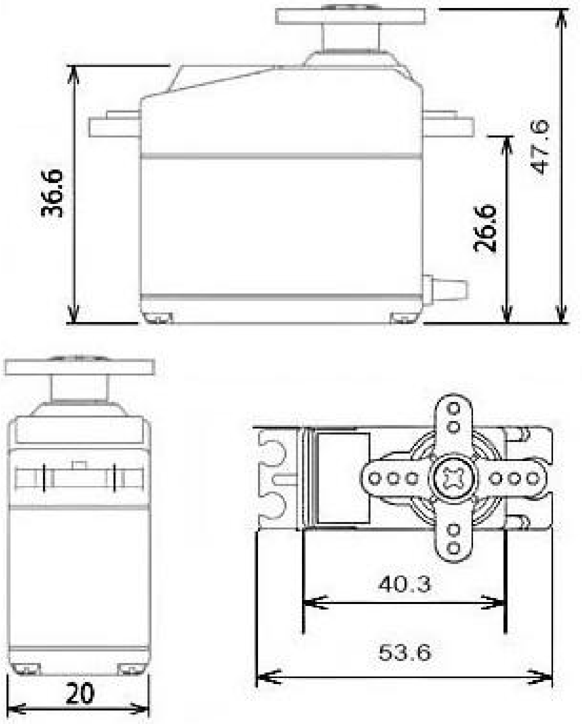
[18] N. S. Nise, *CONTROL SYSTEMS ENGINEERING, (With CD)*. John Wiley & Sons, 2007.

Appendices

Appendix A

Servo motor (MG996R)

MG996R High Torque
Metal Gear Dual Ball Bearing Servo



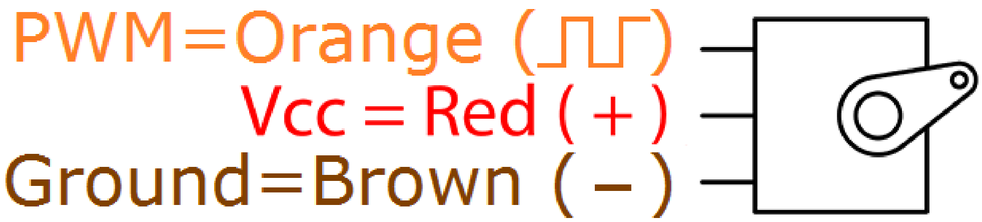
This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwith and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

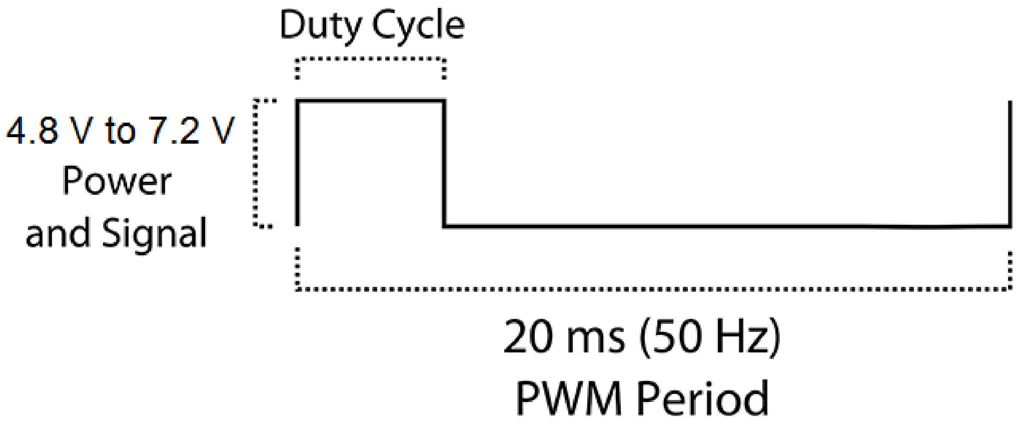
This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60º (4.8 V), 0.14 s/60º (6 V)

- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Temperature range: 0 ºC – 55 ºC





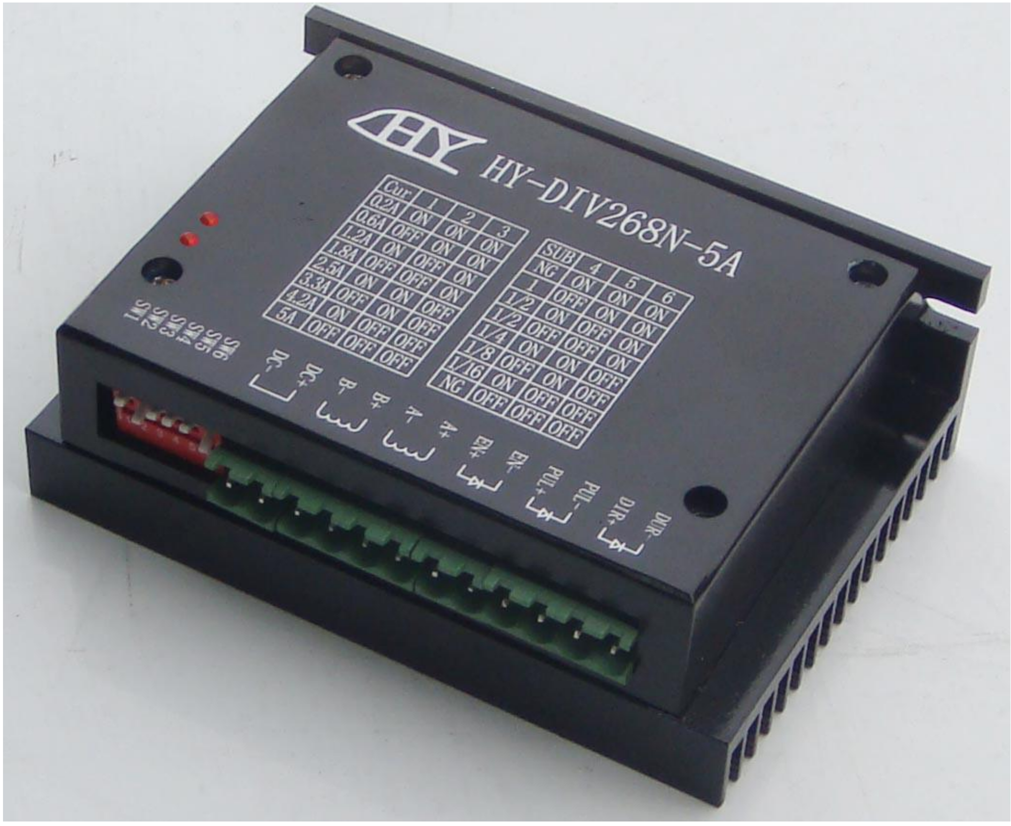
Appendix B

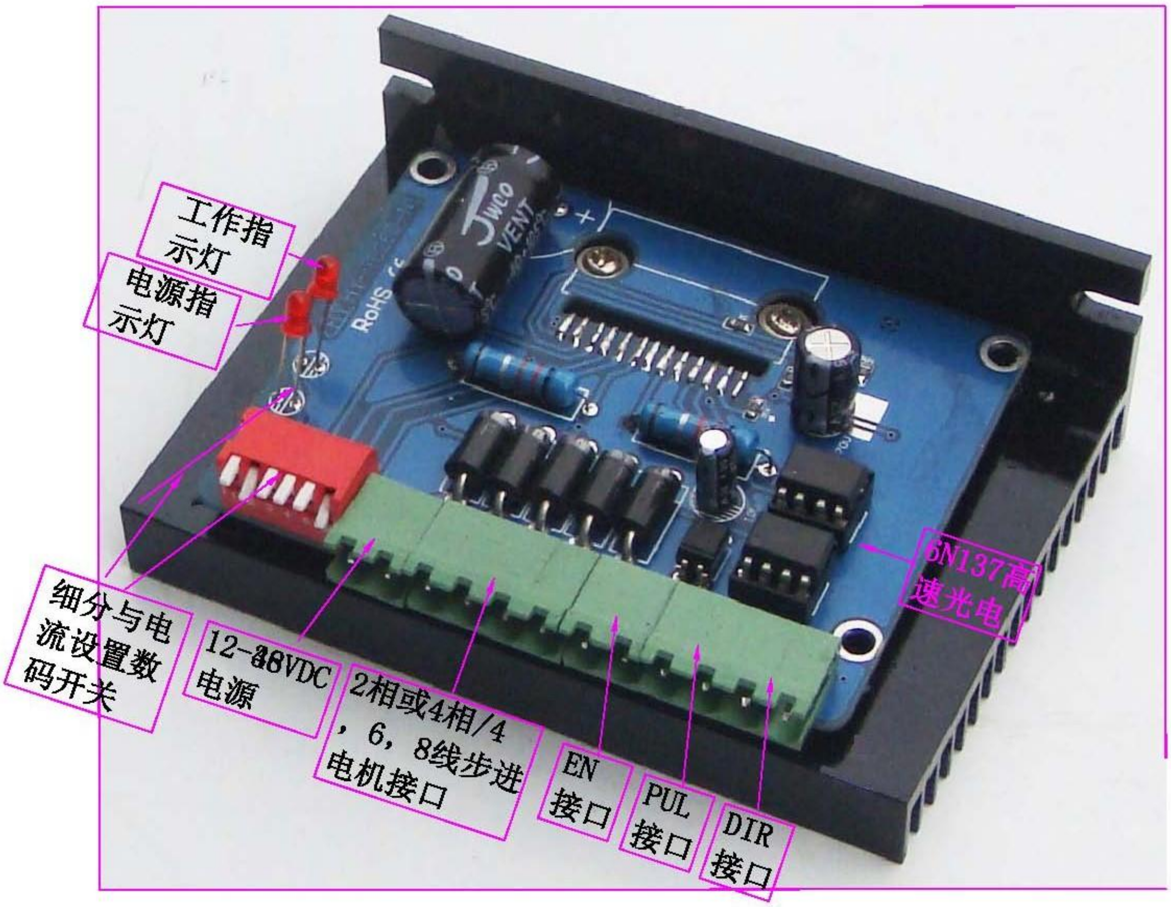
Stepper motor driver (HY-DIV268N-5A)

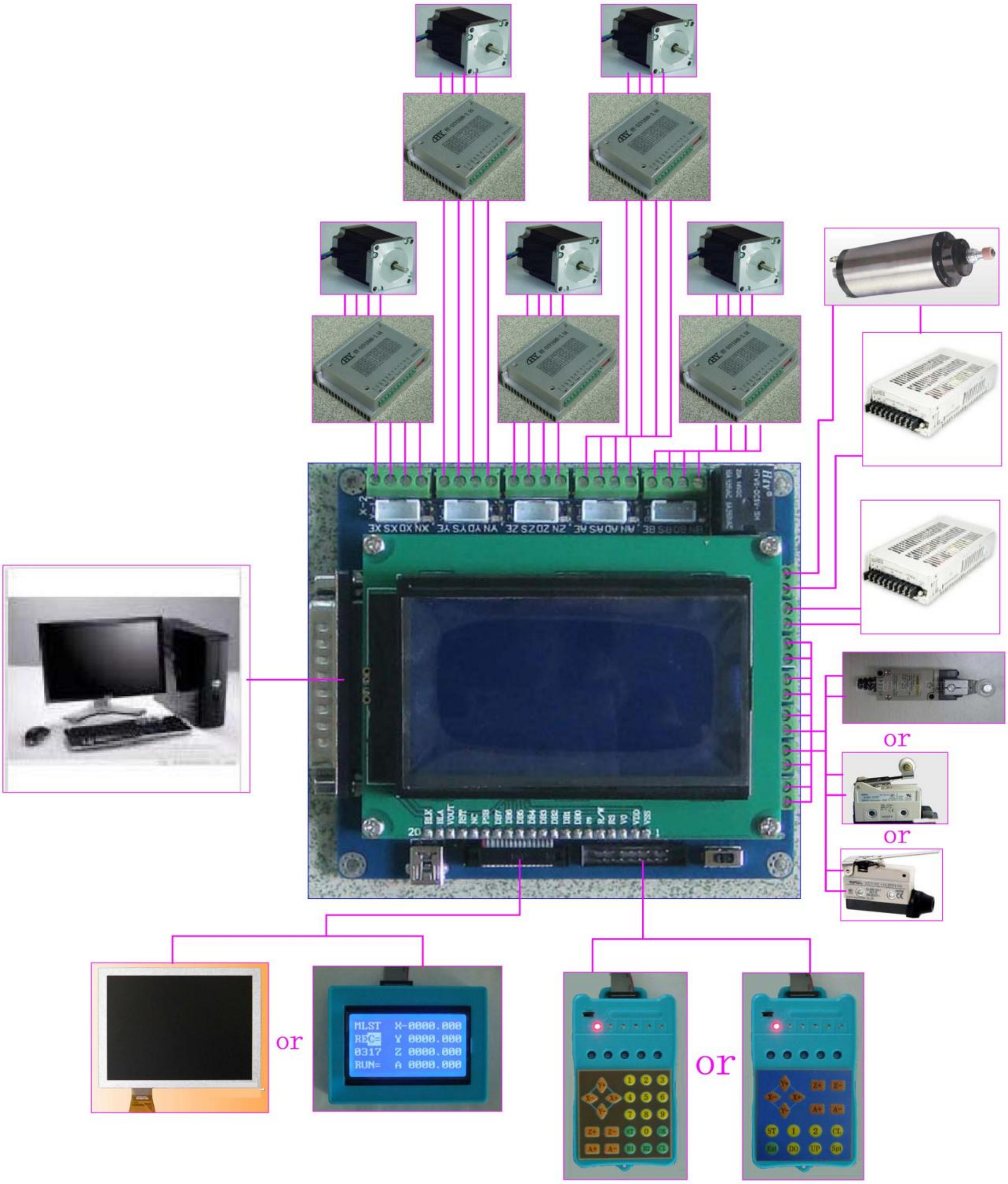
HY-DIV268N-5A two phase hybrid stepper motor drive manual

Product Link:

http://www.thanksbuyer.com/cnc-single-axis-tb6600-0-2-5a-two-phase-hybrid-stepper-motor-driver-controller-24891







一, an overview

HY-DIV268N-5A subdivision-type two-phase hybrid stepping motor drive using DC 12 ~ 48V power supply, suitable for drive

Two-phase hybrid stepping motor dynamic voltage 12 to 48V, the current is less than 5A outer diameter of 35 to 86 mm. This drive using the drive's current loop subdivision control, the motor torque ripple is very small, low-speed running is very smooth, almost no vibration and noise. High-speed torque is much higher than other two-phase drive, high positioning accuracy. Widely used in the engraving machine, CNC machine tools, packaging machinery and other high resolution requirements on the device.

The main features

1 average current control, two-phase sinusoidal current drive output

2 DC 12 ~ 48V power supply, the internal integration of 12V and 5V regulator

3 optically isolated signal input / output

4, overvoltage, undervoltage, overcurrent, and white short-circuit protection

5 4 file segmentation and automatic half-streaming capabilities up to 16 segments
4 file output phase current settings
7 high starting speed
8 high-speed torque
一, the electrical parameters
Input voltage DC 12 ~ 48V input
The input current of 1 to 5 amps, select the drive a stepper motor.
Output current of 0.2A ~ 5A
Temperature Operating Temperature -10 to 45 ℃; Storage temperature -40 ℃ to 70 ℃
Humidity not condensing, not drops
Gas prohibit combustible gas and conductive dust
Weight 200 grams
二、Control signal interface
Figure 1 is a wiring schematic of the drive
1, the definition of control signals
PUL +: step pulse signal is input side or the positive pulse signal input positive terminal
PUL-: the negative input of the negative input pulse signal or a positive pulse signal
DIR +: stepping direction signal input to the positive terminal or negative pulse signal input to the positive terminal
DIR -: stepping direction signal input of the negative side or reverse step pulse signal input negative terminal
EN +: offline can reset signal input side is
EN-: offline can reset signal input negative terminal
Offline enable signal is active, reset drive failure to prohibit any pulse, the output of the drive
Power component is turned off, the motor holding torque.
2、Control signal connections
PC control signal can be high, also can be low effective. When active high, the control signal
The negative side together as a signal to active low, positive side of all control signals together as a signal common.
For example, open-collector and PNP output interface circuit diagram is as follows:

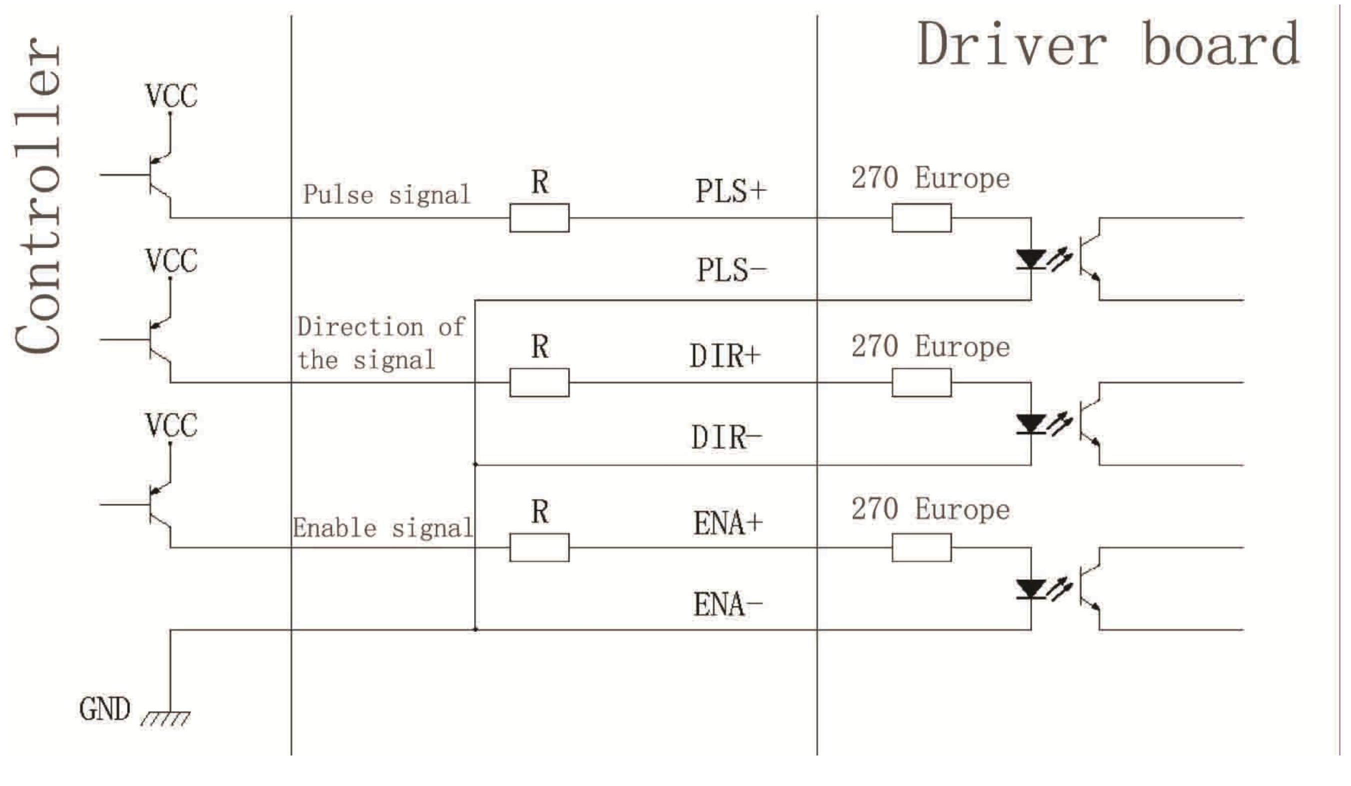


Figure 2. Input interface circuit (common cathode connection)

Note: the VCC is 5V, R short；

VCC value of 12V, R 1K, more than 1/8W resistor;

VCC value of 24V, R 2K, more than 1/8W resistor;

R **Must be connected to the controller output terminals.**

三、Function selection (DIP switch on the drive panel)

1、Set the motor per revolution steps

The drive to set the number of steps per motor revolution is 200 (whole step), 400 (2 segments), 1600 (8 segments), 3200 steps (16 segments).

The user can drive the front panel DIP switch SW3 is SW4 bit of the drive to set the number of steps (such as Table 1）：

Table 1

2、Set the output phase current

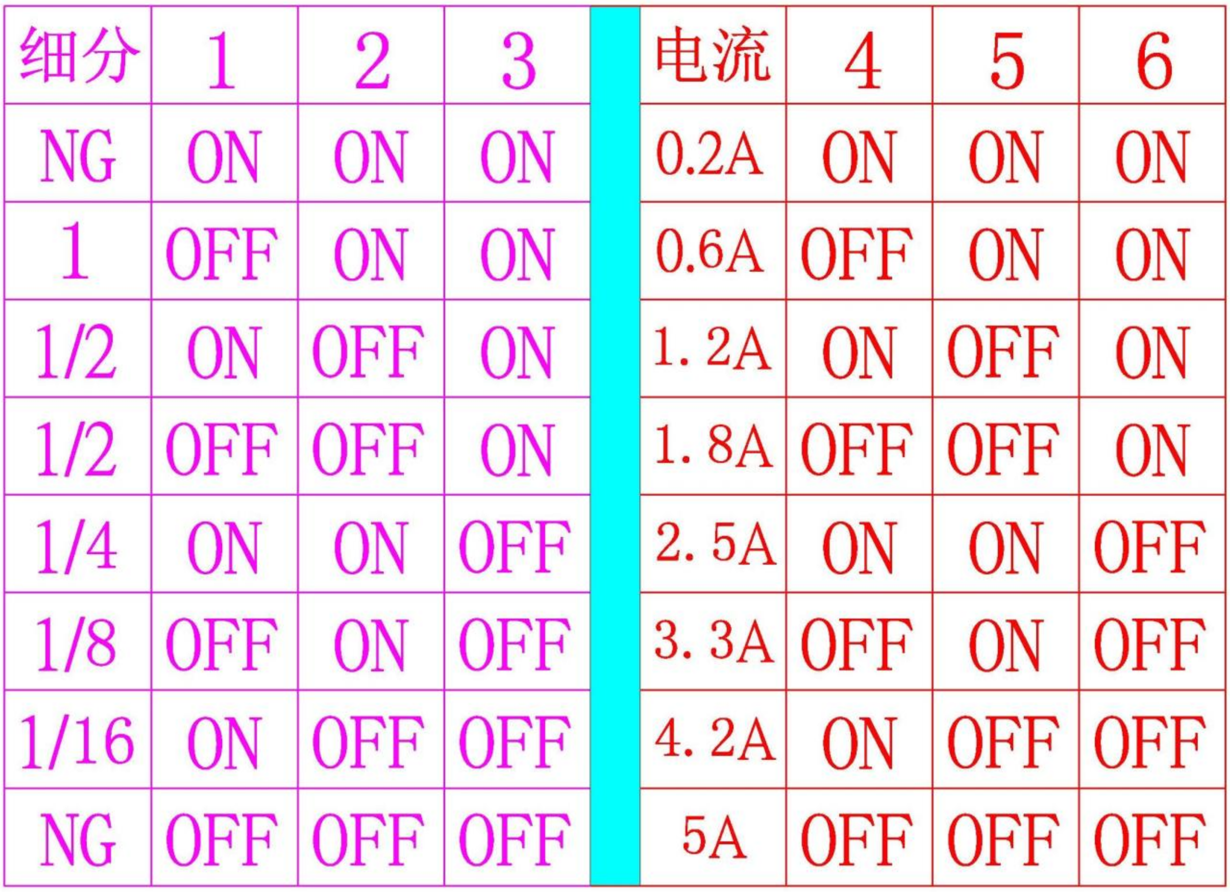
To drive torque stepper motor, the user can drive panel DIP switch

SW1, SW2 is to set the drive output phase current (RMS) unit amperes, the switch position Corresponding to the output current, output current value corresponding different types of drive. Concrete are shown in Table 2.

Output Current (A

Table 2

3、Current setting



细分	1	2	3		电流	4	5	6
NG	ON	ON	ON		0.2A	ON	ON	ON
1	OFF	ON	ON		0.6A	OFF	ON	ON
1/2	ON	OFF	ON		1. 2A	ON	OFF	ON
1/2	OFF	OFF	ON		1. 8A	OFF	OFF	ON
1/4	ON	ON	OFF		2. 5A	ON	ON	OFF
1/8	OFF	ON	OFF		3. 3A	OFF	ON	OFF
1/16	ON	OFF	OFF		4. 2A	ON	OFF	OFF
NG	OFF	OFF	OFF		5A	OFF	OFF	OFF

4, the semi-flow functionality

The semi-flow function is a step pulse 200ms, the driver output current is automatically reduced to the rated output current

50%, used to prevent motor heating.

Fourth, the power interface

1, DC +, DC-: to connect the drive power

DC +: DC power level, power supply voltage DC 12 to 48V. The maximum current is 5A. The

DC-: DC power supply negative level.

2, A + A-B + B-: to connect the two-phase hybrid stepping motor

Drive and two-phase hybrid stepper motor's connection with the four-wire system, the motor windings in parallel and series connection, and connection method, high-speed performance, but the drive current is large (for motor winding current of 1.73 times), Connected in series when the drive current is equal to the motor winding current.

Five, the installation

Around to have 20mm of space, can not be placed next to the other heating equipment, to avoid dust, oil mist,

Corrosive gases, high humidity is too big and strong vibration. +

Figure 3.

Six fault diagnosis

1, the status light indicates

RUN: green light in the normal work.

ERR: red light, power, light, power indicator

Appendix C

MATLAB m-file

```
%The standerd S-Function tamplet written by MathWorks team
% the function rewritten to represent SCARA Robot without linearization
%for the sake of simulation & controller design

function [sys,x0,str,ts,simStateCompliance] = SCARArobot(t,x,u,flag)

switch flag,

  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  case 0,
    [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;

  %%%%%%%%%%%%%%%
  % Derivatives %
  %%%%%%%%%%%%%%%
  case 1,
    sys=mdlDerivatives(t,x,u);

  %%%%%%%%%%
  % Update %
  %%%%%%%%%%
  case 2,
    sys=mdlUpdate(t,x,u);

  %%%%%%%%%%%
  % Outputs %
  %%%%%%%%%%%
  case 3,
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%%%%%%
  % GetTimeOfNextVarHit %
  %%%%%%%%%%%%%%%%%%%%%%%
  case 4,
    sys=mdlGetTimeOfNextVarHit(t,x,u);

  %%%%%%%%%%%%%
  % Terminate %
  %%%%%%%%%%%%%
  case 9,
    sys=mdlTerminate(t,x,u);

  %%%%%%%%%%%%%%%%%%%%
  % Unexpected flags %
  %%%%%%%%%%%%%%%%%%%%
  otherwise
    DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));

end

% end sfuntmpl
```

```
%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
%
% Note that in this example, the values are hard coded.  This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
%
sizes = simsizes;

sizes.NumContStates  = 8;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;
sizes.NumInputs      = 4;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;   % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0  = [0 0 0 0 0 0 0 0];

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts  = [0 0];

% Specify the block simStateCompliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'DisallowSimState' < Error out when saving or restoring the model sim state
simStateCompliance = 'UnknownSimState';

% end mdlInitializeSizes

%
%=============================================================================
% mdlDerivatives
% Return the derivatives for the continuous states.
%=============================================================================
%
function sys=mdlDerivatives(t,x,u)

sys = [];
%parameters (masses lengthes)
ml1=0.71;               %ml1=.4;
ml2=.2335;                     %ml2=.24;
ml34=.5;                       %ml34= .3;

l1=.1172;
l2=.119;
a1=.25;
a2=.25;
J1=0.0053 % J1=.02;
J2=.0014;      % J2=.01;
J4=1.321*10^-5;
%states
%x(1)=q1;        x(3)=q2;        x(5)=q3;        x(7)=q4;
%x(2)=qd1;       x(4)=qd2;       x(6)=qd3;        x(8)=qd4;
c2=cos(x(3));
s2=sin(x(3));
A=-ml2*l2*a1*s2-ml34*a1*a2*s2;
%mass matrix
m11=ml1* l1^2+ ml2 *(l2^2+a1^2+2*l2 *a1* c2 )+...
    ml34* (a2^2+a1^2+2*a1* a2* c2 )+ J1+ J2+J4;
m12=ml2 *(l2^2+l2* a1* c2 )+ ml34* (a2^2+a2* a1* c2 )+ J2+ J4;
m22=ml2* l2^2+ml34* a2^2+J2+J4;
m21=m12;
m14=-J4;m41=-J4;m24=-J4;m42=-J4;m44=J4;
m33=ml34;
M=[m11 m12 0 m14;m21 m22 0 m24;...
      0 0 m33 0; m41 m42 0 m44];
 %coriolos & centrifugal
CC=[A*x(4) A*(x(2)+x(4)) 0 0;-A*x(2) 0 0 0;
          0 0 0 0;0 0 0 0];
  %gravity vector
  gf=[0 0 -ml34*9.8 0]';
% end mdlDerivatives
A=inv(M)*(u-(CC*[x(2) x(4) x(6) x(8)]'+gf));
x1dot=x(2);
x2dot=A(1);
x3dot=x(4);
x4dot=A(2);
x5dot=x(6);
x6dot=A(3);
x7dot=x(8);
x8dot=A(4);
sys=[x1dot x2dot x3dot x4dot x5dot x6dot x7dot x8dot];
%
%=============================================================================
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=============================================================================
%
function sys=mdlUpdate(t,x,u)

sys = [];

% end mdlUpdate

%s
%=============================================================================
% mdlOutputs
% Return the block outputs.
%=============================================================================
%
function sys=mdlOutputs(t,x,u)

sys = [x(2) x(4) x(6) x(8)];

% end mdlOutputs

%
%=============================================================================
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block.  Note that the result is
% absolute time.  Note that this function is only used when you specify a
% variable discrete-time sample time [-2 0] in the sample time array in
% mdlInitializeSizes.
%=============================================================================
%
function sys=mdlGetTimeOfNextVarHit(t,x,u)

sampleTime = 1;    %  Example, set the next hit to be one second later.
sys = t + sampleTime;

% end mdlGetTimeOfNextVarHit

%
%=============================================================================
% mdlTerminate
% Perform any end of simulation tasks.
%=============================================================================
%
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate
```