DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

KINEMATIC ANALYSIS AND IMPLEMENTATIONS OF A ROBOT MANIPULATOR FOR EDUCATIONAL PURPOSES

by

Egemen BAYRAKDAR

March, 2021

İZMİR

KINEMATIC ANALYSIS AND IMPLEMENTATIONS OF A ROBOT MANIPULATOR FOR EDUCATIONAL PURPOSES

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechatronics Engineering, Mechatronics Engineering Program

by
Egemen BAYRAKDAR

March, 2021
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **“KINEMATIC ANALYSIS AND IMPLEMENTATIONS OF A ROBOT MANIPULATOR FOR EDUCATIONAL PURPOSES”** completed by **EGEMEN BAYRAKDAR** under supervision of **ASST. PROF. DR. AYTAÇ GÖREN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Aytaç GÖREN

Supervisor

Asst. Prof. Dr. Özgür TAMER

(Jury Member)

Assoc. Prof. Dr. Yener TAŞKIN

(Jury Member)

Prof. Dr. Özgür ÖZÇELİK

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to express my deepest gratitude my advisor Asst. Prof. Dr. Aytaç GÖREN for his motivating talks, academic guidance, valued support and endless patience throughout the all steps of this thesis.

I am very grateful to my father, Rahmi BAYRAKDAR, my mother, Ümmü BAYRAKDAR, my older brother, Özgür BAYRAKDAR and his lovely wife Gizem BAYRAKDAR for their help, support and love. I could not have completed this work without their constant encouragement and belief.

Finally, I want to emphasize and appreciate moral support of my dear friend Ata POURFARIVARNEZHAD and my friend that experiences similar process while I do too, Merve KARPUZ.

Egemen BAYRAKDAR

KINEMATIC ANALYSIS AND IMPLEMENTATIONS OF A ROBOT MANIPULATOR FOR EDUCATIONAL PURPOSES

ABSTRACT

Nowadays the new industrial age, comes with new focus areas like industry 4.0, robotic manufacturing and internet of things. As it has always been, supply and demand relationships are determining the direction of technological improvements. Hence, robotics and robotic applications have become one of the main focuses of the academics, researchers and students.

Unfortunately, most of the experimental setups of this area, are relatively expensive. Therefore, affordability also became an issue in such matters. It especially applies to educational purposes.

This study examines that whether students, researchers and academics can study with relatively low-cost setup by presenting an example study on a robot arm.

Not all students, researchers and academics can have access to robotic setup all the time. Although a scenario like lockdown situation due to corona virus pandemic is an extreme but valid example, it is not the only one. Infrastructure and access limitations have always been a potential obstacle. Because of that, this study also examines that whether specific two arm scenarios can be calculated with limited one arm preliminary data.

Keywords: Robot, manipulator, two arms, kinematics

EĞİTİM AMAÇLI BİR ROBOT MANİPÜLATÖRÜNÜN KİNEMATİK ANALİZİ VE UYGULAMALARI

ÖZ

Bugünlerde yeni endüstriyel çağ, beraberinde endüstri 4.0, robotik imalat ve nesnelerin interneti gibi odak alanlarını getiriyor. Her zaman olduğu gibi, arz ve talep ilişkileri teknolojik gelişmelerin yönünü belirliyor. Dolayısıyla robotik ve robotik uygulamalar akademisyenler, araştırmacılar ve öğrencilerin ana odaklarından biri haline gelmiştir.

Ne yazık ki bu alandaki birçok deney düzeneği göreceli olarak pahalıdır. Bu yüzden bu konularda maliyet uygunluğu da bir mesele haline gelmiştir. Bu özellikle eğitim amaçları için geçerlidir.

Bu çalışma öğrencilerin, araştırmacıların ve akademisyenlerin göreceli olarak düşük maliyetli düzenek ile çalışıp çalışamayacağını bir robot kol üzerindeki örnek çalışma ile sunarak incelemektedir.

Tüm öğrencilerin, araştırmacıların ve akademisyenlerin her zaman robotik düzeneğe erişimi olmayabilir. Korona virüs pandemisinden dolayı sokağa çıkma kısıtlaması gibi uç bir senaryo da geçerli olsa bile, tek geçerli senaryo değildir. Altyapı ve erişim kısıtlamaları her zaman potansiyel bir engel olmuştur. Bu sebeple bu çalışma, sınırlı tek kol ön verisi ile belirli iki kol senaryolarının hesaplanabilmesini de incelemektedir.

Anahtar kelimeler: Robot, manipülatör, iki kol, kinematik

CONTENTS

Sayfa

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

Due to their manufacturing advantages, robotic manipulators are attracting interest of researchers, students and academics. Although there are many well qualified models in the market, these relatively expensive model catalogues like ABB (2021) and Kuka (2021) started to raise the question of affordability for researchers and enthusiasts.

Uicker, Pennock & Shigley (2011) historically explain that “The term *robot* is of Slavic origin, comes from the Czech word *robota*, meaning work or labor, and was introduced into the English vocabulary by Czechoslovakian dramatist Karl Capek in the early part of the 20th century.” and goes on with mentioning computer and electromechanical aspects of it. Because robot is compilation of multiple components. Stadtler (1995) points out that there are seven essential components of a robot. These are alphabetically actuator, communicator, control computer, end effector, manipulator, power supply and sensor.

TQ MA 2000 is the name of the robot manipulator model that was used as a basis in this study. This manipulator is the product of TecQuipment International Ltd and manufactured due to need of robotic arm at Britain’s Open University due to educational purposes in early 1980’s (Beebcontrol, 2021).

Context of the manual for this robot manipulator was helpful for understanding working limitations of the model. The reach of the robot is 500 mm, that presents working space limited to this distance. Force calculations bound to be related with end effector attachment point. Based on this point, distance is 480 mm. This is the length that should be used in force calculations. The force limitations are given with two different values. First one is 1 kg dead lift value, which is simpler and less dynamic weight endurance for the manipulator. Second value is 500 g limitation for more dynamic action of lifting and moving objects (TecQuipment, n.d.).

According to Tsai (1999), robots can be classified in 5 different criteria. These classification criterions are degrees of freedom, kinematic structure, drive technology, workspace geometry and motion characteristics. By the same order, TQ MA 2000 can be classified as a 6 degree of freedom, serial, electric drive, articulated, spatial robot manipulator.

Educational robotic studies are very common in the field. Paunski & Angelov (2019) presented a cost effectivity of an educational robot by using cheaper material and optimizing the power consumption, Manzoor et al.(2014) did another study with another perspective by using open source educational platform. Since educational robotics are not only for academics, but also for kids, there are lot of social studies about the topic like Ospennikova et al. (2015). All of these studies backed up or created richer infrastructure for more technical studies like trajectory planning of Liu X. et al. (2015), Cruz et al. (2018) and also dual arm studies like Buhl et al. (2019).

As a case study for two arm applications, this study examines two commonly studied example actions of twisting and folding. Besides lifting applications of dual arm robotics like obstacle application of Yang et al. (2019), folding applications like robotic origami study of Phuong et al. (2018) or a task of carton folding study like Liu H. et al. (2003) are also commonly studied. Task of twisting or screwing is another common application of dual arm task that can be seen in many studies like Shauri (2012) that presents sensor integration point of view or assembly application of Krüger (2011). As an addition, these applications are not only limited to assembly applications. There are different studies like surgical application of Wu (2019).

In this study, further information about experimental setup and measurements are presented in chapter two for a better understanding. Which is followed by Chapter three, that includes kinematic calculation and applications. Chapter four demonstrates an investigation of two arm scenarios that extrapolated from chapter three. After four chapters, chapter five presents closing arguments of this study.

CHAPTER TWO
EXPERIMENTAL SETUP AND MEASUREMENTS

2.1 Experimental Setup

Tecquipment’s TQ MA 2000 model robot arm was used for this study as mentioned in chapter one. Reviving a used, second hand TQ MA 2000, required some renovation endeavors. Despite of its age, this robot arm had a good mechanical foundation.

Figure 2.1 TQ MA 2000 robot manipulator (Personal archive, 2020)

In 1980's this robot manipulator originally designed to be controlled by a big sized control box which did not allow external software override. Only buttons on the box could control the robot. Therefore, it was resistant to modern day microcontrollers and embedded pc like Arduino and Raspberry Pi. Because of this compatibility issue two main cable lines, which connected robot and control box, were examined for creating alternative control options. After identifying obvious connections, others determined by trial and error with multimeter measurements and related supply voltages.

This kind of experimental discovery was necessary due to insufficient information at the manual of the arm. Some parameters from manual were given in chapter one. Although a lot of information on the manual has a little use for this study due to most information on the manual being reliant on its original control box, parameters about system constraints were very useful. Figure 2.2 presents angular limitation of six axes.

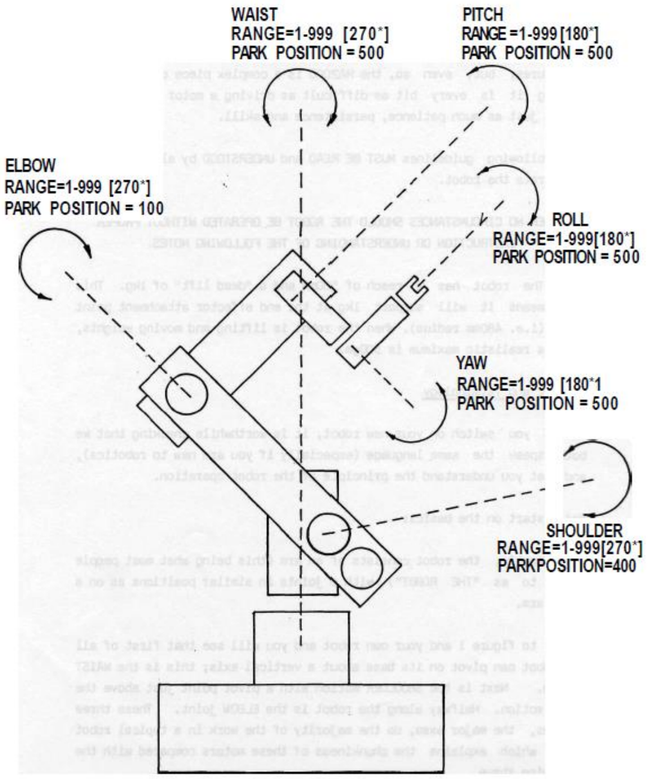

Figure 2.2 Major and minor axes showing park position (Tequipment, n.d.)

2.1.1 Exploration of Unknown Connections and Data

Figure 2.3 TQ MA 2000 Sensor Connection Socket (Personal archive, 2020)

One of these cable lines, which shown in Figure 2.3, was determined as a sensor line. Electrical resistance change due to robot arm movements, measured by a multimeter. Hence, rotational position sensors identified as potentiometer. After this realization every binary combination was tested until the full identification of the cable line. There are 6 analog data carriers (1-2-2-4-5-6) and 3 pairs of voltage supply lines (7-8-9-10-11-12).

Figure 2.4 TQ MA 2000 Actuator Connection Socket (Personal archive, 2020)

The other cable line, that shown in Figure 2.4 was identified as actuator line. Movement in robot arm gave voltage readings in some binary combination of sockets (3-4, 7-8, 11-12, 15-16, 19-20, 23-24) without any power source. Therefore, motors of the manipulator identified as electrical direct current motors. Rest of the sockets were identified as pneumatic socket outlets for valve of gripper (1-5-9-13-17-21).

Contents of the manual for this robot has very few information about dimensional measurements. For modeling purposes and better understanding of the system, some of the dimensions were measured. These measurements, which are shown from Figure 2.5 to Figure 2.13. Lengths up to 150 mm were measured by DDC61232F model Mac Allister brand digital Caliper. Which has 0.01 mm graduation, +/-0.02 mm sensitivity when shorter than 100 mm, +/- 0.03 mm sensitivity when longer than

100 mm and working temperatures in range of 5-40 Celsius degrees. Lengths that are higher than 150 mm were measured more roughly by 1-30-497 model Stanley brand tape measure. (Mac Tools, 2021)

Figure 2.5 Measurements illustrated at the base of TQ MA 2000 (Personal archive, 2020)

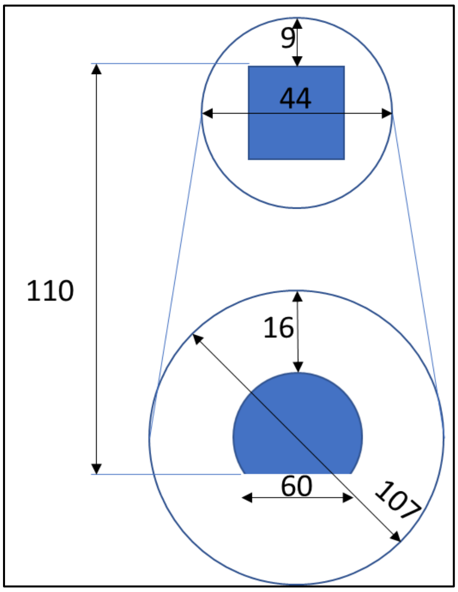

Figure 2.6 Measurements of the cam that shown at Figure 2.5

Figure 2.7 Different angle of the basis that demonstrate further measurements (Personal archive, 2020)

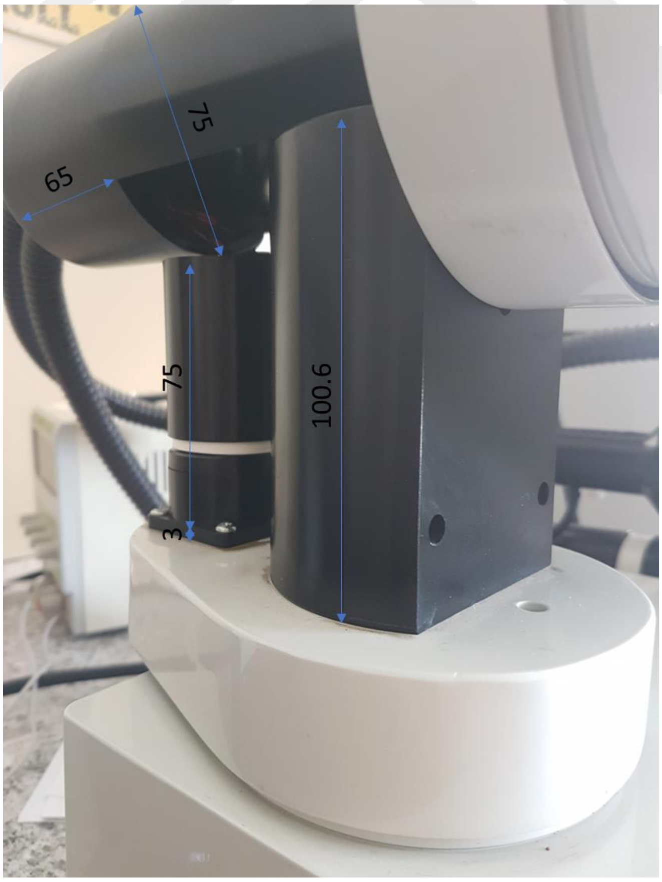

Figure 2.8 Measurements between first two actuators from the base (Personal archive, 2020)

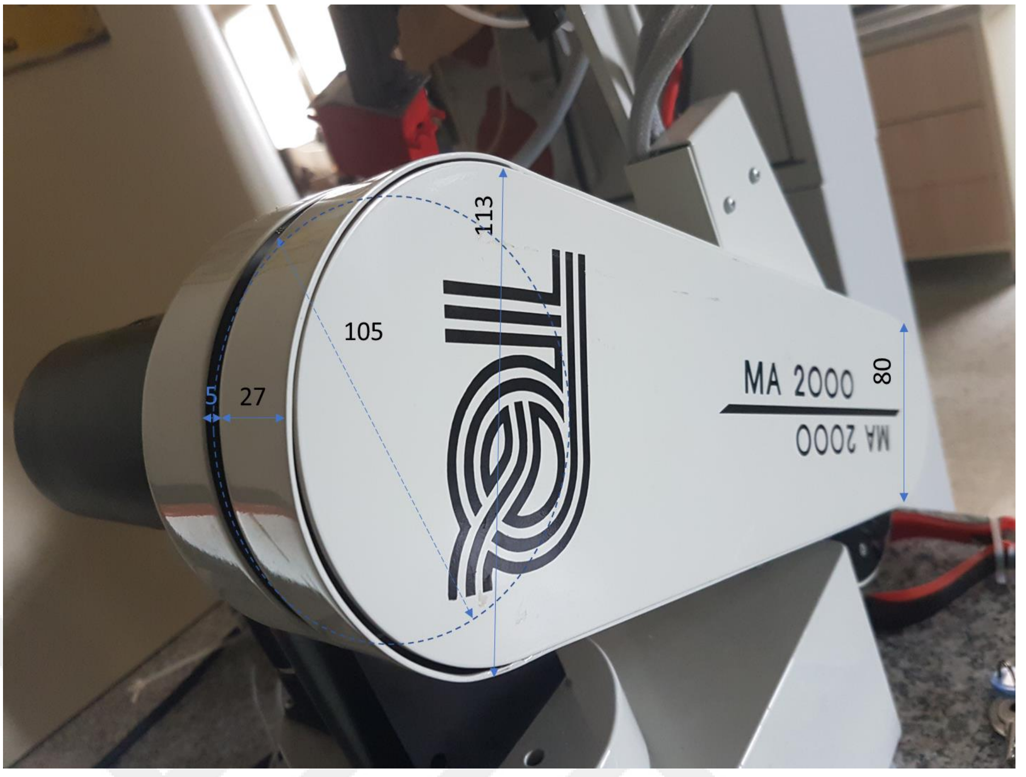

Figure 2.9 Measurements between second and third actuators from the base (Personal archive, 2020)

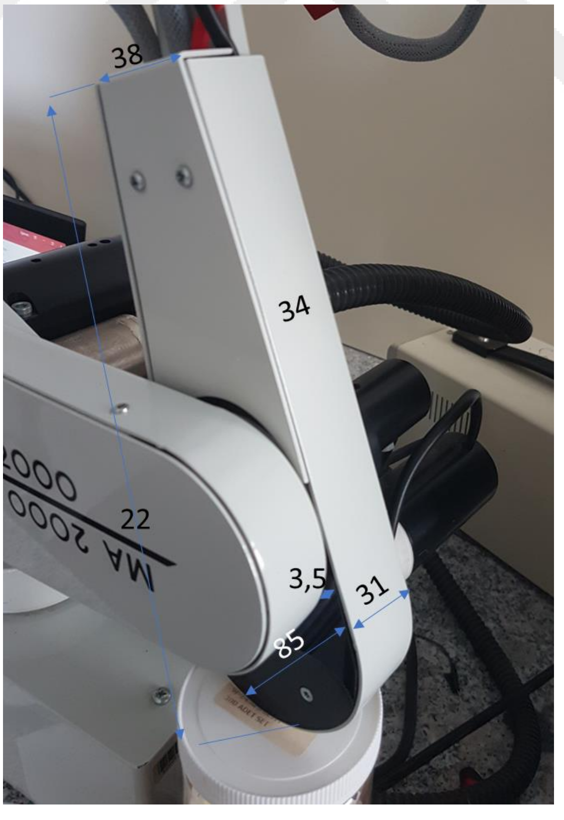

Figure 2.10 Front view measurements between third and fourth actuator from the base (Personal archive, 2020)

Figure 2.11 Back view measurements between third and fourth actuator from the base (Personal archive, 2020)

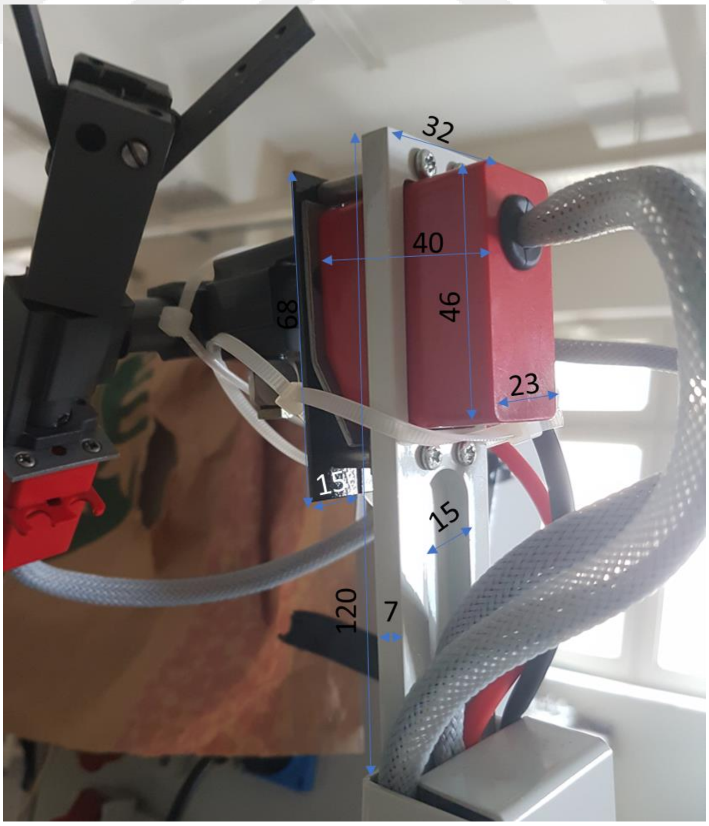

Figure 2.12 Wrist measurements (Personal archive, 2020)

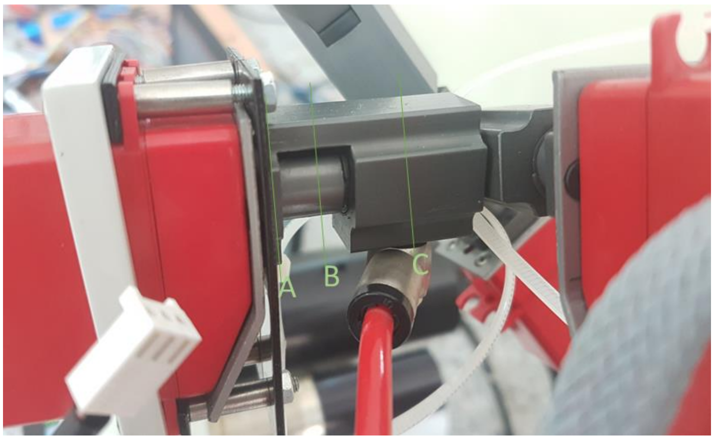

Figure 2.13 Wrist with addressing of 3 cross sections as A,B and C (Personal archive, 2020)

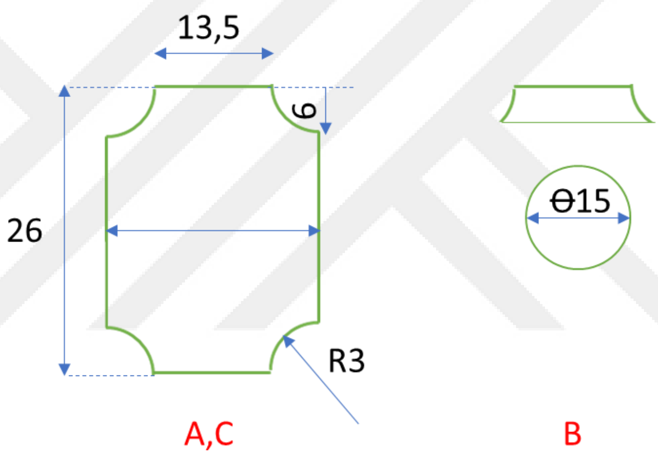

Figure 2.14 Technical drawing representation of cross sections that were addressed at Figure 2.13

Distances between rotational axes are also essential dimensions. These dimensions have an important role in kinematics. Because of that, they are represented in following figure and table and will be used in upcoming chapters.

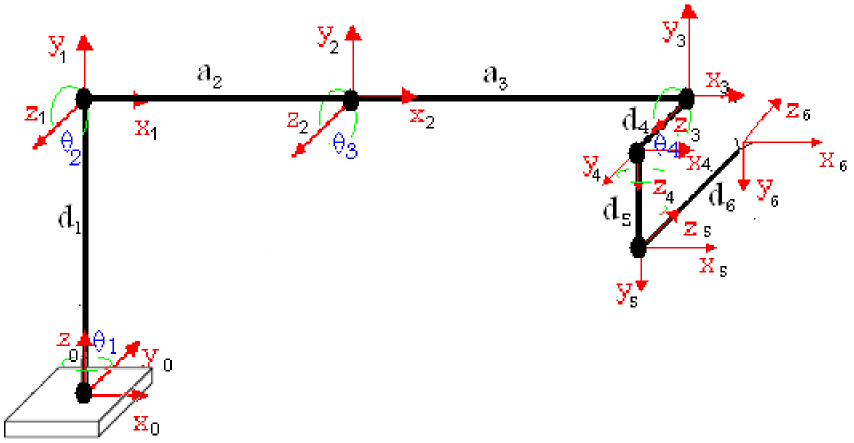

Figure 2.15 Graphical representation of TQ MA 2000 (Al-Faiz & Saleh, 2011)

Table 2.1 Distance values based on parameter definitions of Figure 2.15

i	a_i (cm)	d_i (cm)
1	0	26
2	23	0
3	24	0
4	0	5
5	0	4.4
6	0	8

2.1.2 System and Derived data

As mentioned in chapter 1, for a system to be recognized as a robot, there must be 7 essential components. TQ MA 2000 robot manipulator can not work as a stand-alone system. This chapter focuses on these matters and searches uses of previous data.

2.1.2.1 CAD Model

All dimensions from subchapter 2.1.2, can be used to create a 3D model of TQ MA 2000 robot manipulator. Therefore, a model like in the Figure below created by using SolidWorks software.

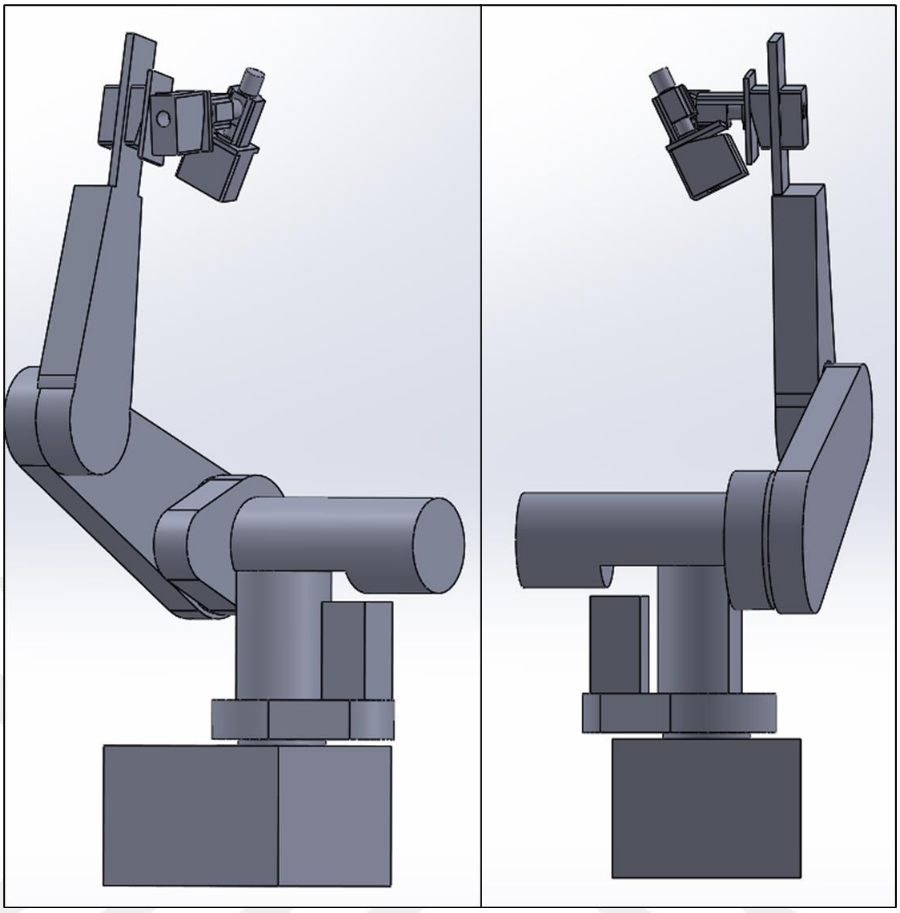

Figure 2.16 3D SolidWorks drawing of TQ MA2000 from 2 opposite angles

2.1.2.2 Renovation of Sensor Connections

Since the original control box of TQ MA 2000 is closed black box system, sensor interpretations should be discovered for adapting robot to modern age control systems. Sensors of the manipulator were identified as potentiometers in subchapter 2.1.1 and sockets shown of Figure 2.3 separated to 12 different cable lines.

Same process was also applied to actuator line and multi-cable connections for the controller prepared. That is shown below.

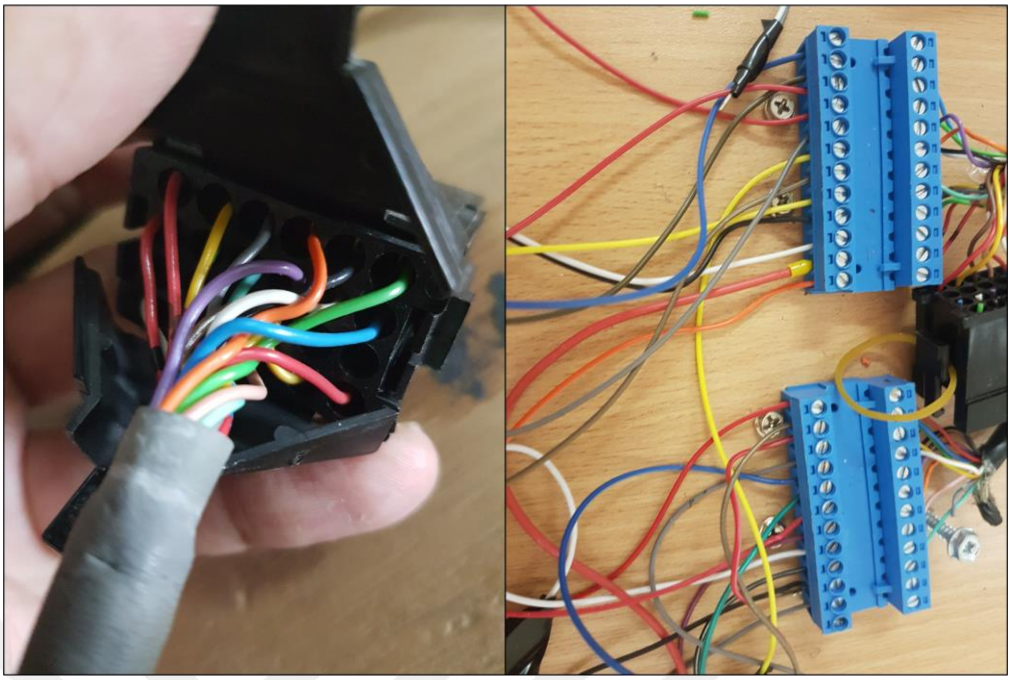

Figure 2.17 Before-after illustration of cable separation (Personal archive, 2020)

This study used Arduino Mega model of Arduino microcontroller. Because of that, renovations made for its capabilities. But similar adaptations can be made for other controller of choice. Arduino Mega operates with 5 Volt of output voltage. Hence, sensor curves desired for 5V voltage.

From this point, actuators and sensors starting from base to end effector will be numbered in order as shown in Figure 2.18. For example, second sensor or sensor no 2 refers to second potentiometer from base to end effector.

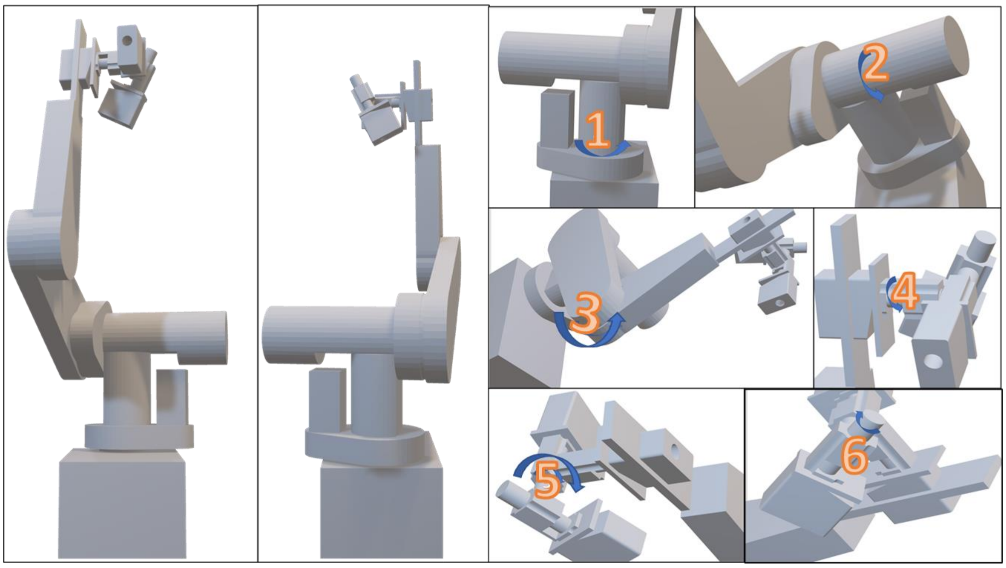

Figure 2.18 Illustration of sensor and actuator numeration on 3D Model

Potentiometers are known for having linear relation between position and resistance. This relation is commonly used to determine the calibration curve of the potentiometer. Because of that, at least 10 samples of analog reading values (between 0 – 1023) were measured with 5V voltage supply of Arduino in different angular positions. At following 6 tables, results of 6 potentiometers and their consecutive voltage differences can be found. The average of these differences was used to determine slope of the linear sensor calibration equation and measured position values were reference positions to the ground.

Table 2.2 Voltage Measurements of Potentiometer 1

<table>
<tr><td colspan="5">Potentiometer 1</td></tr>
<tr><td>Position</td><td>Analog Input</td><td>Voltage (V)</td><td></td><td></td></tr>
<tr><td>0</td><td>770</td><td>3.763</td><td>__</td><td>ΔV</td></tr>
<tr><td>10</td><td>753</td><td>3.680</td><td>___|----></td><td>0.083</td></tr>
<tr><td>20</td><td>733</td><td>3.583</td><td>___|----></td><td>0.098</td></tr>
<tr><td>30</td><td>715</td><td>3.495</td><td>___|----></td><td>0.088</td></tr>
<tr><td>40</td><td>693</td><td>3.387</td><td>___|----></td><td>0.108</td></tr>
<tr><td>50</td><td>677</td><td>3.309</td><td>___|----></td><td>0.078</td></tr>
<tr><td>60</td><td>656</td><td>3.206</td><td>___|----></td><td>0.103</td></tr>
<tr><td>70</td><td>639</td><td>3.123</td><td>___|----></td><td>0.083</td></tr>
<tr><td>80</td><td>620</td><td>3.030</td><td>___|----></td><td>0.093</td></tr>
<tr><td>90</td><td>602</td><td>2.942</td><td>___|----></td><td>0.088</td></tr>
<tr><td>100</td><td>583</td><td>2.849</td><td>___|----></td><td>0.093</td></tr>
<tr><td>110</td><td>566</td><td>2.766</td><td>___|----></td><td>0.083</td></tr>
<tr><td></td><td></td><td colspan="2">AverageΔV:</td><td>0.091</td></tr>
</table>

Table 2.3 Voltage Measurements of Potentiometer 2

<table>
<tr><td colspan="5">Potentiometer 2</td></tr>
<tr><td>Position</td><td>Analog Input</td><td>Voltage (V)</td><td></td><td></td></tr>
<tr><td>0</td><td>674</td><td>3.294</td><td>___</td><td>ΔV</td></tr>
<tr><td>10</td><td>656</td><td>3.206</td><td>___|----></td><td>0.088</td></tr>
<tr><td>20</td><td>637</td><td>3.113</td><td>___|----></td><td>0.093</td></tr>
<tr><td>30</td><td>616</td><td>3.011</td><td>___|----></td><td>0.103</td></tr>
<tr><td>40</td><td>599</td><td>2.928</td><td>___|----></td><td>0.083</td></tr>
<tr><td>50</td><td>577</td><td>2.820</td><td>___|----></td><td>0.108</td></tr>
<tr><td>60</td><td>561</td><td>2.742</td><td>___|----></td><td>0.078</td></tr>
<tr><td>70</td><td>543</td><td>2.654</td><td>___|----></td><td>0.088</td></tr>
<tr><td>80</td><td>523</td><td>2.556</td><td>___|----></td><td>0.098</td></tr>
<tr><td>90</td><td>506</td><td>2.473</td><td>___|----></td><td>0.083</td></tr>
<tr><td>100</td><td>487</td><td>2.380</td><td>___|----></td><td>0.093</td></tr>
<tr><td>110</td><td>469</td><td>2.292</td><td>___|----></td><td>0.088</td></tr>
<tr><td></td><td></td><td colspan="2">AverageΔV:</td><td>0.091</td></tr>
</table>

Table 2.4 Voltage Measurements of Potentiometer 3

Potentiometer 3				
Position	Analog Input	Voltage (V)		
0	179	0.875	___	ΔV
10	189	0.924	___|---->	0.049
20	197	0.963	___|---->	0.039
30	206	1.007	___|---->	0.044
40	215	1.051	___|---->	0.044
50	225	1.100	___|---->	0.049
60	233	1.139	___|---->	0.039
70	241	1.178	___|---->	0.039
80	251	1.227	___|---->	0.049
90	260	1.271	___|---->	0.044
100	268	1.310	___|---->	0.039
110	274	1.339	___|---->	0.029
120	282	1.378	___|---->	0.039
		AverageΔV:		0.041

Table 2.5 Voltage Measurements of Potentiometer 4

Potentiometer 4				
Position	Analog Input	Voltage (V)		
0	153	0.748	___	ΔV
10	162	0.792	___|---->	0.044
20	171	0.836	___|---->	0.044
30	182	0.890	___|---->	0.054
40	191	0.934	___|---->	0.044
50	202	0.987	___|---->	0.054
60	214	1.046	___|---->	0.059
70	223	1.090	___|---->	0.044
80	234	1.144	___|---->	0.054
90	244	1.193	___|---->	0.049
100	253	1.237	___|---->	0.044
		AverageΔV:		0.049

Table 2.6 Voltage Measurements of Potentiometer 5

Potentiometer 5				
Position	Analog Input	Voltage (V)		
0	115	0.562	___	ΔV
10	123	0.601	___|---->	0.039
20	132	0.645	___|---->	0.044
30	143	0.699	___|---->	0.054
40	153	0.748	___|---->	0.049
50	165	0.806	___|---->	0.059
60	176	0.860	___|---->	0.054
70	185	0.904	___|---->	0.044
80	197	0.963	___|---->	0.059
90	206	1.007	___|---->	0.044
100	214	1.046	___|---->	0.039
		AverageΔV:		0.048

Table 2.7 Voltage Measurements of Potentiometer 6

Potentiometer 6				
Position	Analog Input	Voltage (V)		
0	216	1.056	___	ΔV
10	225	1.100	___|---->	0.044
20	233	1.139	___|---->	0.039
30	243	1.188	___|---->	0.049
40	254	1.241	___|---->	0.054
50	266	1.300	___|---->	0.059
60	277	1.354	___|---->	0.054
70	286	1.398	___|---->	0.044
80	298	1.457	___|---->	0.059
90	308	1.505	___|---->	0.049
100	317	1.549	___|---->	0.044
		AverageΔV:		0.049

Simple equation of a line was given in equation below and potentiometer-angle relation is also known to be a line. 6 linear calibration functions can be defined with previously acquired sensor measurements.

$$y = m(x - x_{ref}) \tag{2.1}$$

Where y is angular position, m is the slope of the equation, x_{ref} is the reference voltage value for zero degrees and x is the input voltage value. Based on that, sensor characteristics at Table 2.8 were found. These characteristics will be used be microcontroller (Arduino) to interpret the angular position data.

Table 2.8 Sensor characteristics

i	ΔV_i	$V_{ref.i}$	m(degree/Volt)
1	0.091	3.763	110.324
2	0.091	3.294	109.785
3	0.042	0.875	238.369
4	0.049	0.748	204.6
5	0.048	0.562	206. 667
6	0.049	1.055	202.574

2.1.2.3 Actuator Speed Profiles

TQ MA 2000 robotic setup has 6 actuators that were identified as dc motor in subchapter 2.1. Among these 6 motors there are 2 different models of dc motor. Due to insufficient information in the manual, speed profiles related to power were unknown. Also working characteristics on the robot arm couldn’t be found in any document. Because of that, these motors were connected to L298 motor driver, and angular position change over time were recorded with different power inputs. The relation between magnitude of L298’s voltage input and motor speed were observed and recorded between 2 intervals. These intervals are from 30 to 80 and from -20 to 200 degrees. Movement of both positive and negative direction have been measured. 30 to 80 degree scale has chosen for having example movement under 90 degrees where center of gravity of link does not cross over other side of the motor. Second set of scale was chosen for observing whether there will be any change when 90 degree position has been passed during the movement.

From now on, movement from 30 degree to 80 degree will be referred as first rise and movement with inverse directions will be referred as first fall. Also, movement from -20 degree to 200 degree will be referred as second rise and reverse of this movement will be referred as second fall. As mentioned, TQ MA 2000 has two different kind of dc motor. Before defining motor speed characteristics, first three actuators from the base will be referred as type A motors and last three actuators from the base will be referred as type B.

As mentioned before, this study investigates angular position over time with different magnitude of L298 motor driver inputs. Inputs applied scale between 0-255. Working on the arm with different voltage supply percentages showed that for motors to work stable, best working range in 0-255 range, is 15-170. Voltage supply ratio is pwm signal generated as an output to enable pin of motor driver circuit, in the range of 0-255. Where 0 is 0 V output and 255 means 5V output to the enable pin of motor. Following pages will refer power supply ratio as V.S.R.

An example calculation presented at figures below with following two consecutive tables. First table gives result of 9 cross calculations of the marked points and second table derives an angular speed from it.

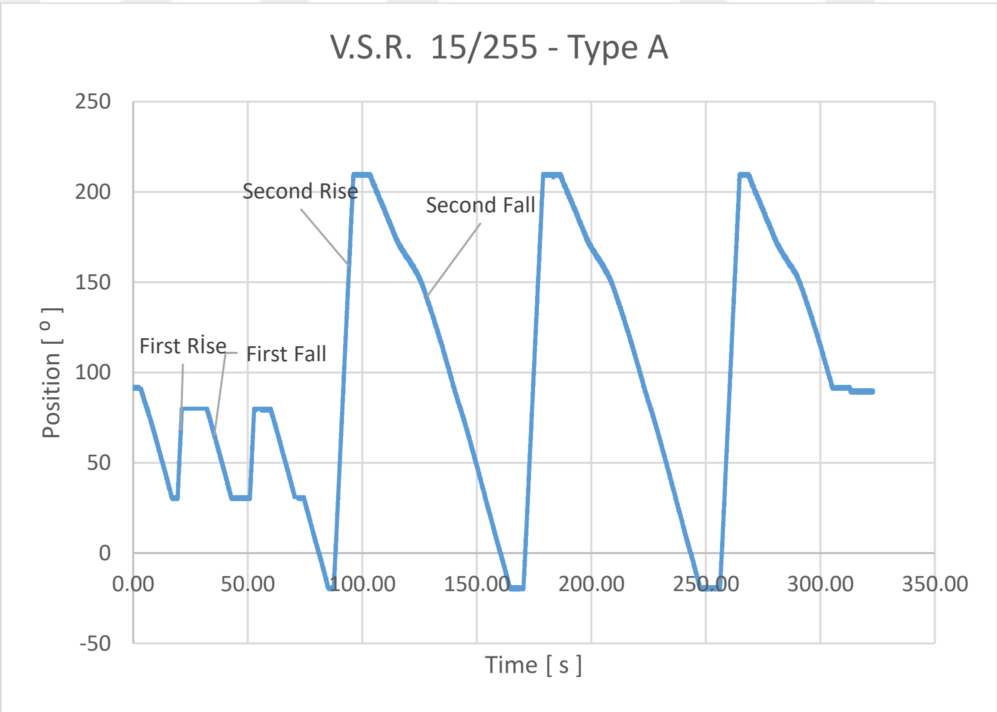

Figure 2.19 Angular position records for type A motor with 15/255 V.S.R.

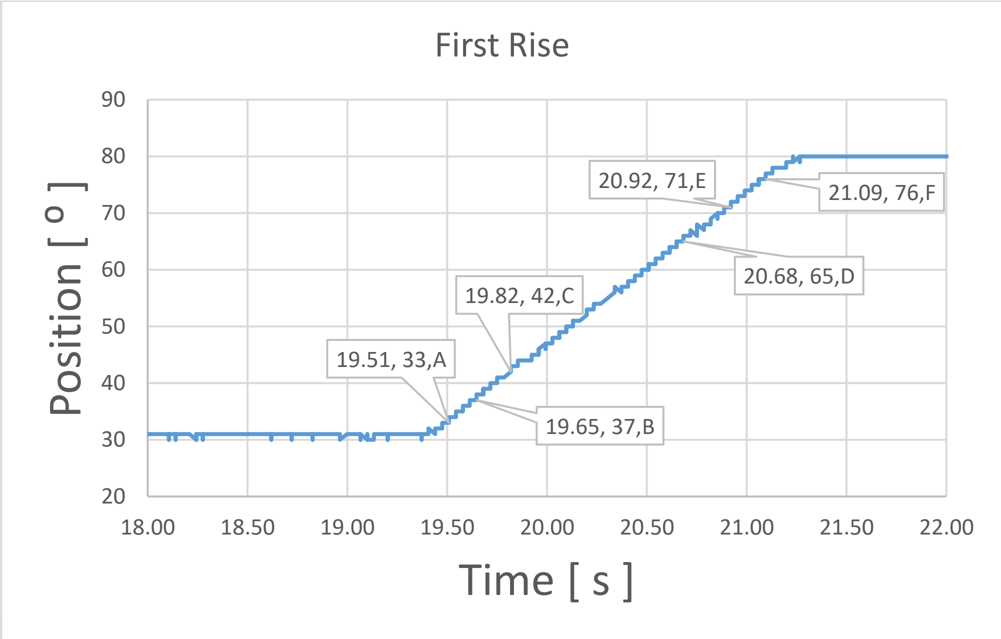

Figure 2.20 A closer look to marked first rise at Figure 2.19

Table 2.9 Cross calculation of slopes that marked in Figure 2.20

Slope	D	E	F
A	27.3504	26.9504	27.2152
B	27.1845	26.7717	27.0833
C	26.7442	26.3636	26.7717

Table 2.10 Angular Speed (Average slope values of Table 2.9) in different units

	deg/s	rad/s	rpm
Av. Slope:	26.9372	0.47014	4.48954

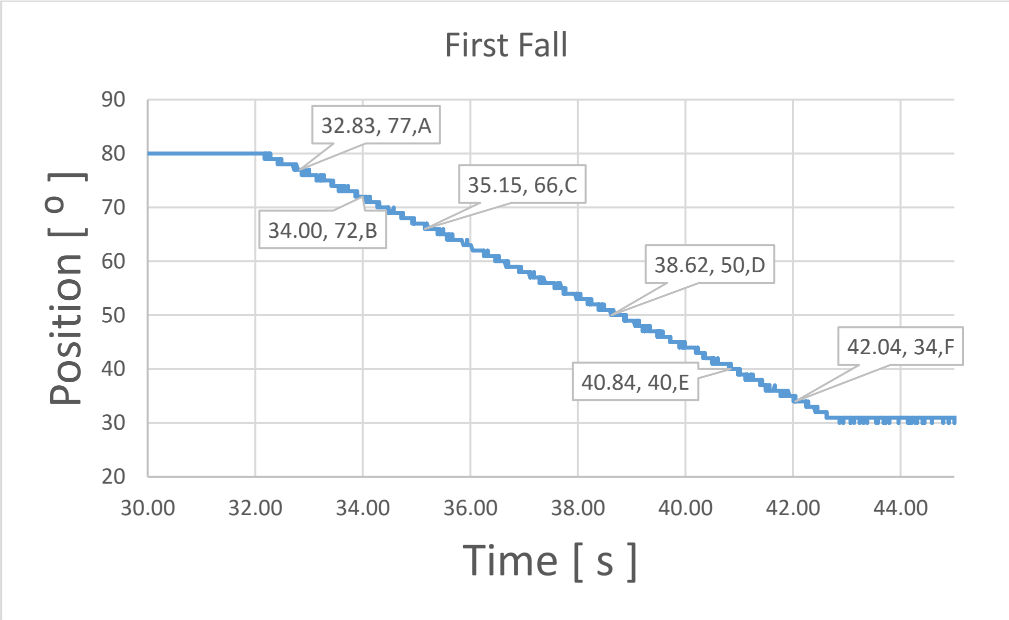

Figure 2.21 A closer look to marked first fall at Figure 2.19

Table 2.11 Cross calculation of slopes that marked in Figure 2.21

Slope	D	E	F
A	-4.6632	-4.6192	-4.6688
B	-4.7619	-4.6784	-4.7264
C	-4.611	-4.5694	-4.6444

Table 2.12 Angular Speed (Average slope values of Table 2.11) in different units

	deg/s	rad/s	rpm
Av. Slope:	-4.6603	-0.0813	-0.7767

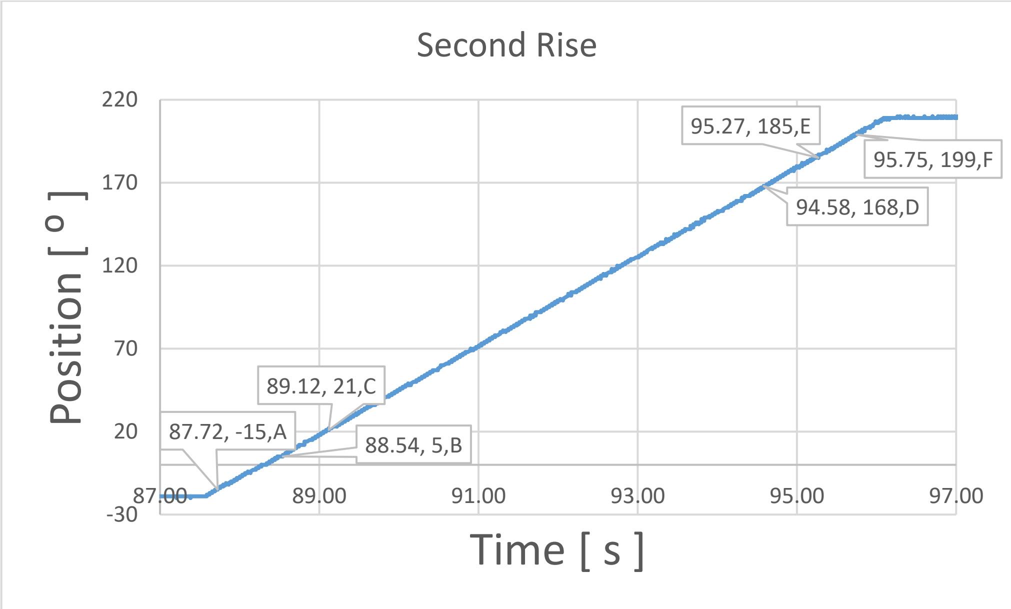

Figure 2.22 A closer look to marked second rise at Figure 2.19

Table 2.13 Cross calculation of slopes that marked in Figure 2.22

Slope	D	E	F
A	26.6764	26.4901	26.6501
B	26.9868	26.7459	26.9071
C	26.9231	26.6667	26.8477

Table 2.14 Angular Speed (Average slope values of Table 2.13) in different units

	deg/s	rad/s	rpm
Av. Slope:	26.766	0.46715	4.46099

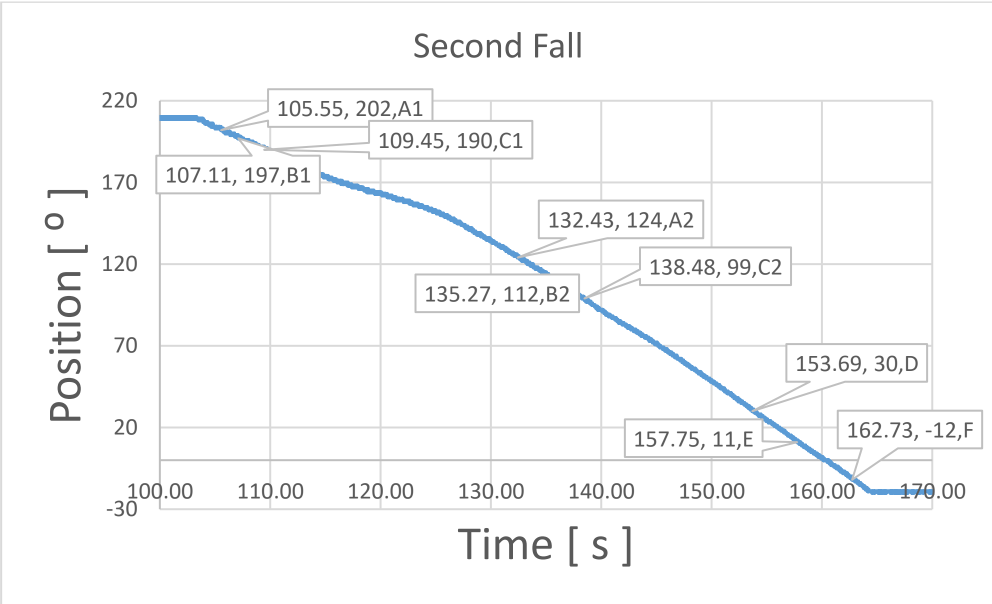

Figure 2.23 A closer look to marked second fall at Figure 2.19

Due to linearity errors in Figure 2.23, points A2, B2 and C2 were considered in calculation. Nevertheless, for observing error in non-linear condition, calculations were also applied to points A1, B1 and C1.

Table 2.15 Cross calculation of slopes that marked in Figure 2.23 for non-linear condition

Slope	D	E	F
A1	-3.5729	-3.659	-3.7426
B1	-3.5852	-3.673	-3.7576
C1	-3.6166	-3.706	-3.7913

Table 2.16 Angular Speed (Average slope values of Table 2.15) in different units

	deg/s	rad/s	rpm
Av. Slope:	-3.6783	-0.0642	-0.613

Table 2.17 Cross calculation of slopes that marked in Figure 2.23 for linear condition

Slope	D	E	F
A2	-4.4214	-4.4629	-4.4884
B2	-4.4517	-4.4929	-4.5157
C2	-4.5365	-4.5667	-4.5773

Table 2.18 Angular Speed (Average slope values of Table 2.17) in different units

	deg/s	rad/s	rpm
Av. Slope:	-4.5015	-0.0786	-0.7502

The same sequence of calculations was made for voltage supply ratios of 15, 20, 30, 40, 50, 80, 90, 100, 120, 150, 170 in the scale of 0-255 for both A and B type motors. These ratios represent the motor driver supply voltages of 0.29V, 0.39V, 0.59V, 0.78V, 0.98V, 1.57V, 1.76V, 1.96V, 2.35V, 2.94V, 3.33V.

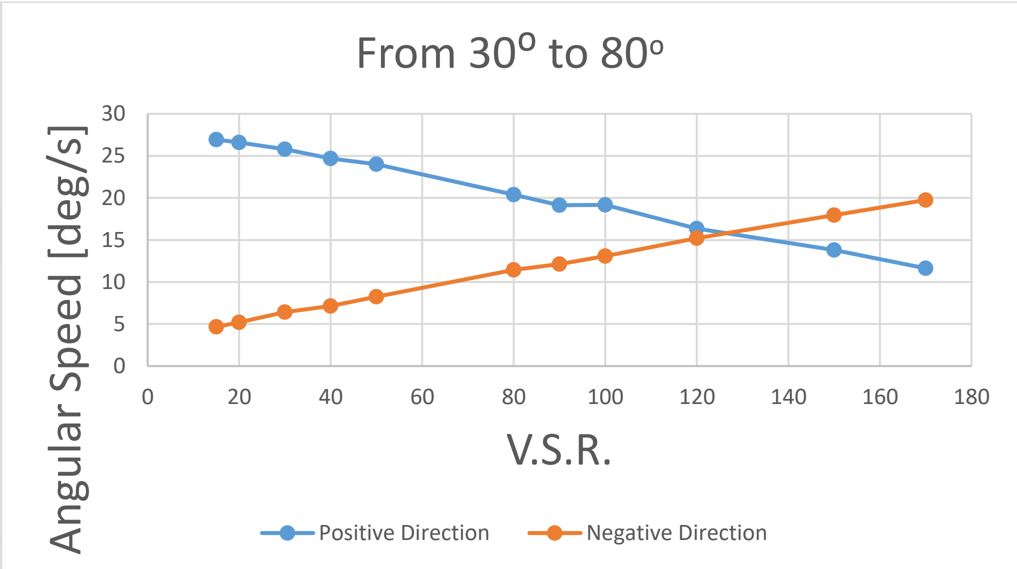

Figure 2.24 Motor control curve due to first rise and fall for motor type A

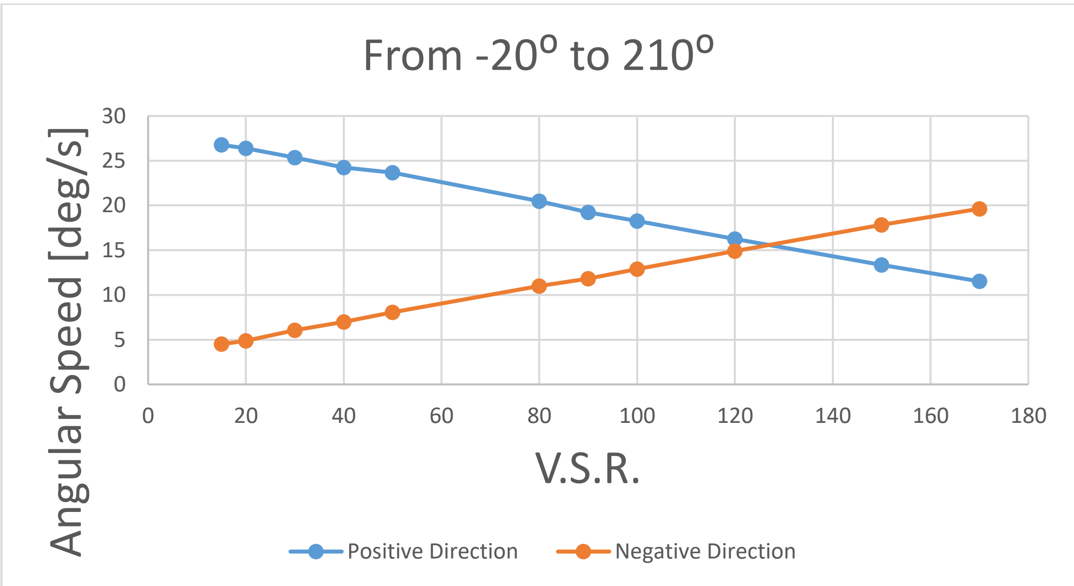

Figure 2.25 Motor control curve due to second rise and fall for motor type A

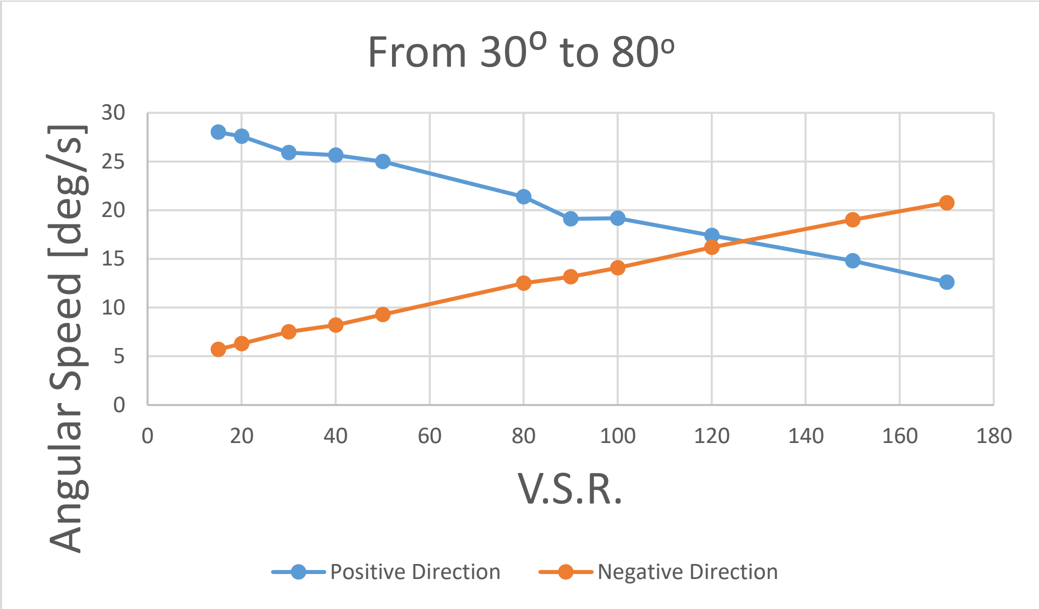

Figure 2.26 Motor control curve due to first rise and fall for motor type B

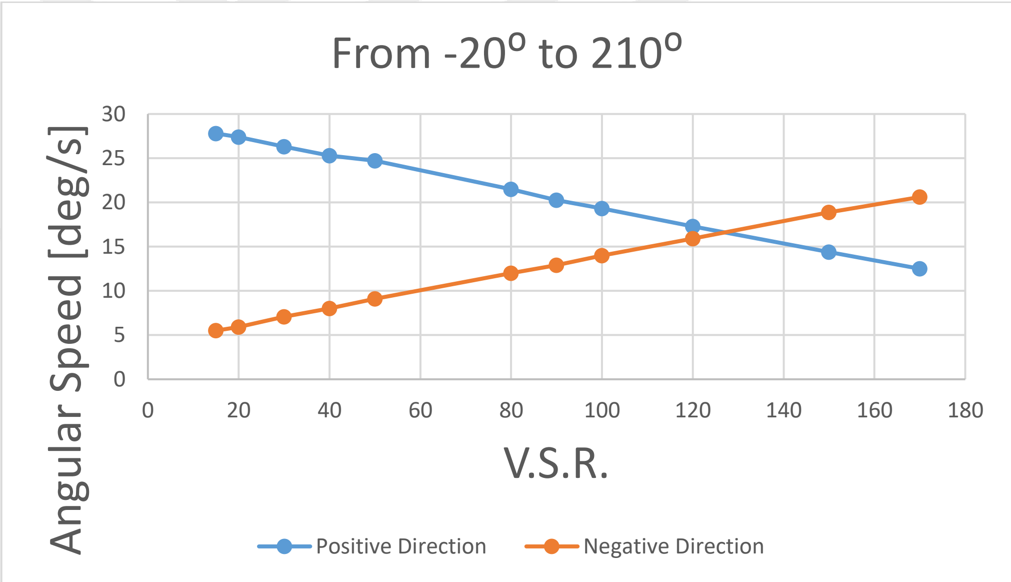

Figure 2.27 Motor control curve due to second rise and fall for motor type B

2.1.2.4 System Connections

Previous chapters explain how to make TQ MA 2000 manipulator compatible with modern day controllers like Arduino. Following figure illustrates a basic explanation of how those connections were made.

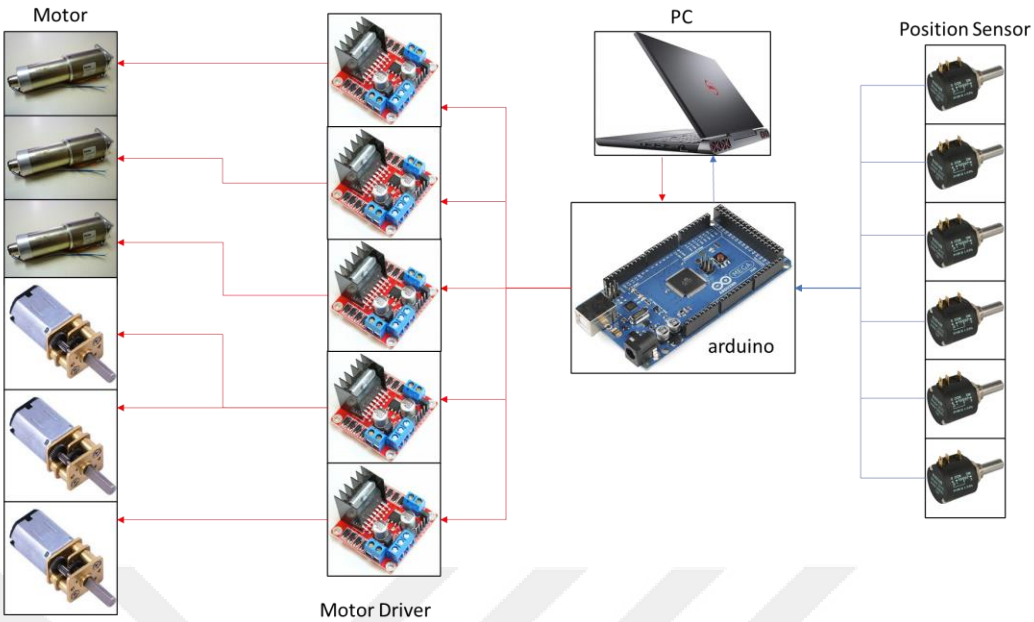

Figure 2.28 System Schematic (Vatan, 2021; Rs, 2021; Ideasoft, 2021; Toolex, 2021; Hepsiburada, 2021)

Before defining sensors and actuators to Arduino, cable connections must be made. Position should be read with analog input from potentiometers and motor should be driven by digital outputs. Arduino connections of motor drivers and potentiometer are shown in following figure.

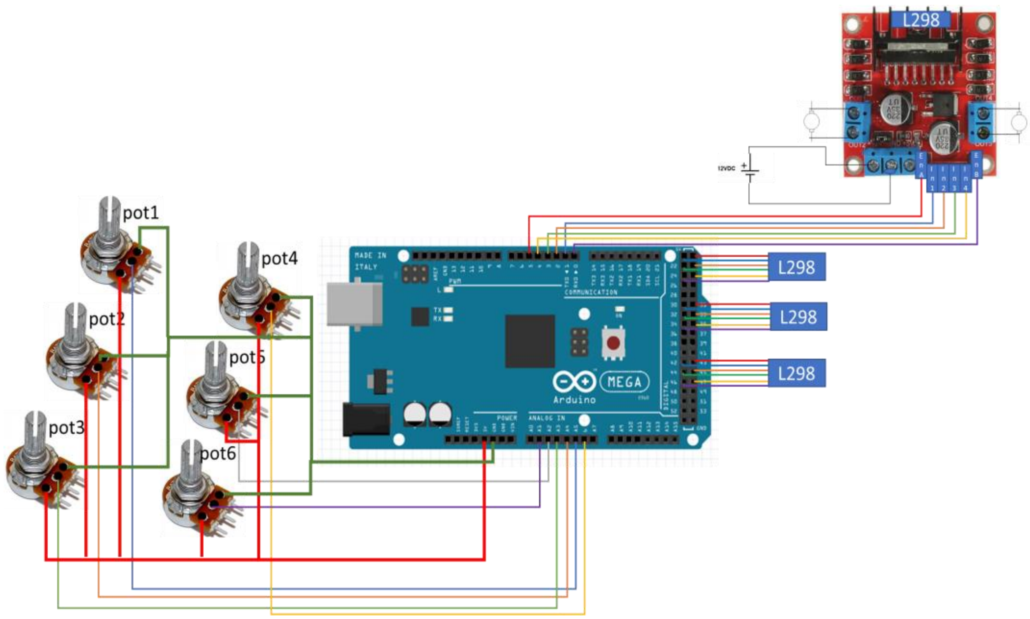

Figure 2.29 Illustration of analog and digital connections of Arduino (Adapted from Robotistan, 2021; Amazon, 2021; Robokits, 2021)

2.2 Software of Arduino and its Definitions

TQ MA 2000 programmed with Arduino as mentioned. Source code was created flexibly. Because of that, a lot of sub program created, therefore software offers a simpler robotic language based on original Arduino’s C++ codes. The full source code can be found in appendix A.

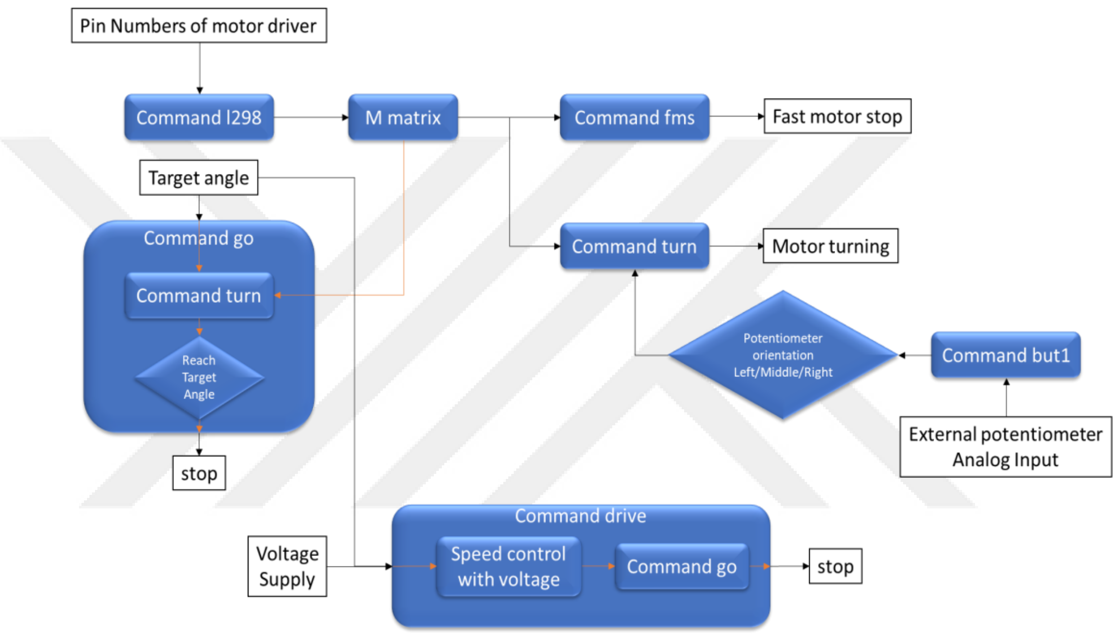

Figure 2.30 Summary of software subprograms

There are 7 subprograms, which created for this software. These are given in following subchapters.

2.2.1 Command l298

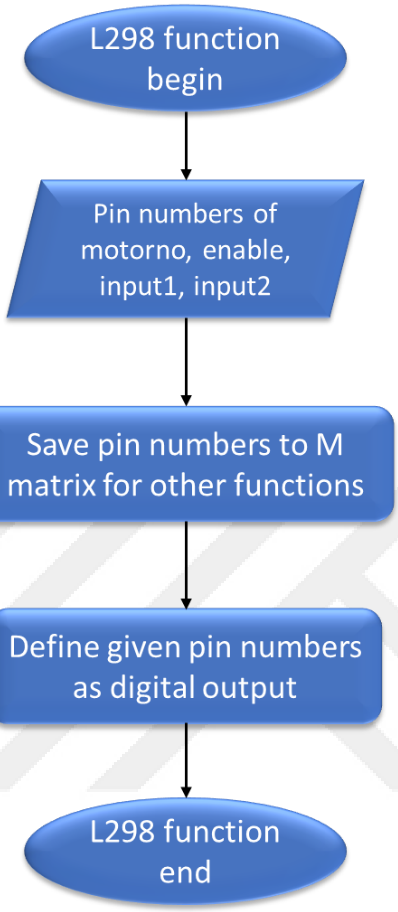

Figure 2.31 Flowchart of l298 command

Command l298 is a command that defines physical connections of motor driver to software with only entering the input of connected pin no.

l298(motorno, enable, input1, input2);

Command addresses *enable* and 2 input pins (*input1* & *input2*) of L298 motor driver to *motorno* numbered motor by saving it to *M* matrix and automatically defines the pins of the driver as digital output. The full content of this program is given below.

void l298(int motorno,int enable, int input1, int input2) { // l298 function that addresses pin numbers of L298 motor controller unit
M[motorno][1]= enable; pinMode(enable,OUTPUT);
M[motorno][2]= input1; pinMode(input1,OUTPUT);
M[motorno][3]= input2; pinMode(input2,OUTPUT);
deneme = enable;}

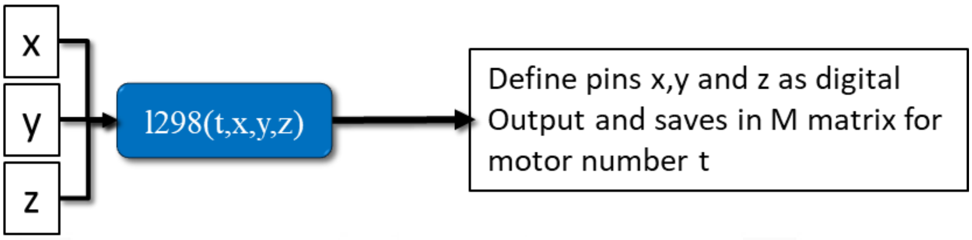

Figure 2.32 Working principle of l298 command

2.2.2 Command interpret

Command interpret is a command that directly takes chosen potentiometer value and processes it with sensor characteristics (m coefficient) that were illustrated in Table 2.8 to an angular output and saves it to *p* vector array.

interpret(potno, m);

This command reads analog data from *potno* numbered potentiometer and subtracts reference zero analog values that previously defined in *refz* vector from it. Then applies *m* coefficient, which previously determined at Table 2.8 and changes analog data to angle. The full content of this program is given below.

void interpret(int potno,int m){ // Interpreting sensor data to position values
*p[potno]= (analogRead(potno), - refz[potno])*5*m/1023 ;*
}

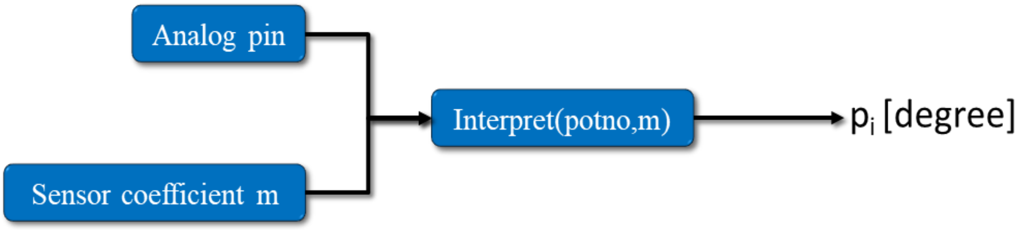

Figure 2.33 Working principle of Interpret command

2.2.3 Command turn

Command turn has a requirement of command l298. Because it uses pin numbers that command l298 defined. It simply turns desired motor in desired speed and direction.

turn(motorno, sped);

This command turns *motorno* numbered motor with the speed of *sped* in the scale of -255 to 255. Below and above zero turns in opposite direction, so that direction and speed can be given with one input parameter. By doing that, this program also checks motor driver from *mctype* vector. So, the code is compatible with different kinds of motor driver. After checking motor driver it turns motor with commanding pins that previously defined in command l298. The full content of this program is given below.

```
void turn(int motorno,  int sped) { //turn function commands related motor drivers to turn motor bidirectionally
if(mctype[motorno]==1)  //Detection of L298 Motor Driver
{
analogWrite(M[motorno][1],sped);     //overriding pwm speed signal with a value between -255 and 255
if(sped<0){digitalWrite(M[motorno][2],HIGH);digitalWrite(M[motorno][3],LOW);
} //positive direction settings for positive speed value
if(sped>0){digitalWrite(M[motorno][2],LOW);digitalWrite(M[motorno][3],HIGH);
} //negative direction settings for negative speed value
  }
```

```
 if(mctype[motorno]==2)  //Detection of MC33926 Motor Driver
 {}

}
```

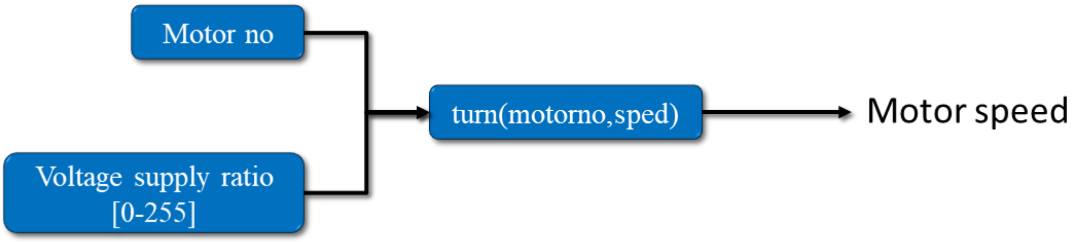

Figure 2.34 Working principle of turn command

2.2.4 Command fms

Name of the command fms is an abbreviation for fat motor stop. Which is one of its functions that mentioned on datasheet of L298 motor driver (sparkfun,2021).

fms(motorno);

It is a very simple command with one input and its function is stopping desired motor, that given as parameter *motorno*, instantly. The full content of this program is given below.

```
void fms(int motorno){  // Fast motor stop program
if(mctype[motorno]=1){digitalWrite(M[motorno][1],HIGH);digitalWrite(M[motorno][2],LOW);digitalWrite(M[motorno][3],LOW);}
}
```

2.2.5 Command but1

Command but1 was developed for testing purposes. Its function is letting an externally connected potentiometer to control desired motor. By this command, when potentiometer position in somewhere middle, motor will not move. When turned to one end, it will turn to a direction and when it turned to reverse end, it’ll turn to opposite direction. Command but1 lets experimenter to actively control a motor without changing software, by using an experimental potentiometer as a button.

but1(motorno, analog);

Command but1 lets experimenter to actively control a motor without changing software, by using an experimental potentiometer as a button. Externally connected potentiometers analog reading is input as *analog* for controlling *motorno* numbered motor. The full content of this program is given below.

```
void but1(int motorno, int analog){  // button control 1 - pot
int c;
if(analog>700) {c=100;}
else if(analog<400) {c=-100;}
else {c=0;}
turn(motorno,c);
}
```

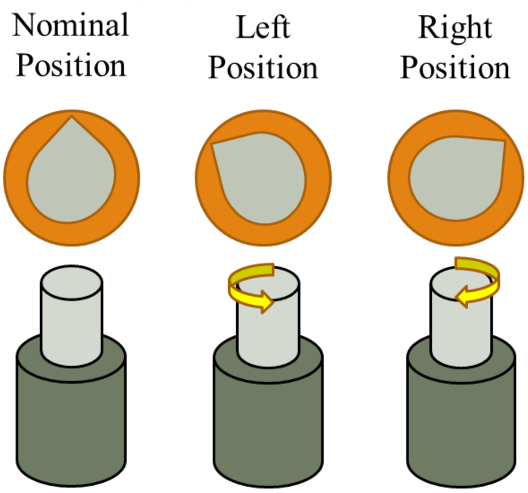

Figure 2.35 Working principle of but1 command

2.2.6 Command go

This command allows desired motor to reach desired angular position. It has a requirement of defining command turn, because it is used in source code.

go(motorno, aci);

When command go applied, *motorno* numbered motor goes to position of *aci* degrees. The full content of this program is given below.

```
void go(int motorno, int aci){  // driving commanded motor to desired position
int hiz;
if(p[2]<(aci-1)){hiz=-100;}
if(p[2]>=(aci-1)&&p[2]<=(aci+1)){hiz=0;}
if(p[2]>(aci+1)){hiz=100;}
turn(motorno,hiz);
}
```

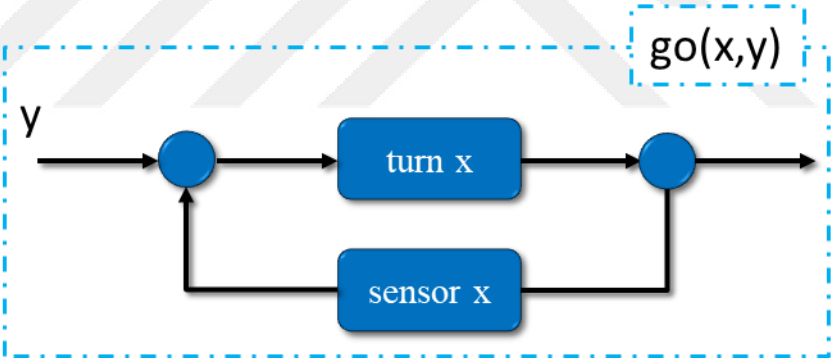

Figure 2.36 Working principle of go command

2.2.7 Command drive

Command drive is very similar to command go. Its only difference is command drive can specify speed of the action.

drive(motorno, aci, hiz);

Its function is making *motorno* numbered motor to reach degree of *aci* with speed *hiz* in the scale of 0-255. Like command go, this program has also requirement of command turn, but unlike command turn, this program does not accept negative speed values. Because the movement direction is automatically determined. The full content of this program is given below.

```
void drive(int motorno, int aci,int hiz){   // driving commanded motor to desired position
if(p[2]<(aci-1)){hiz=-hiz*1;}
if(p[2]>=(aci-1)&&p[2]<=(aci+1)){hiz=hiz*0;}
if(p[2]>(aci+1)){hiz=hiz*1;}
turn(motorno,hiz);
}
```

An example of this command: *drive(2,60,120);*

In this example commands function is bringing motor 2 to the position of 60 degrees with 120/255 of maximum power.

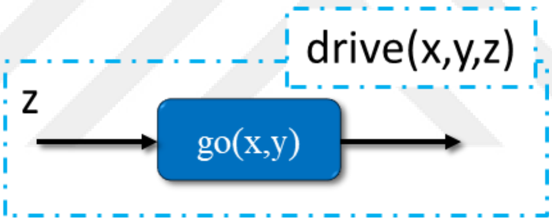

Figure 2.37 Working principle of drive command

CHAPTER THREE
KINEMETIC CALCULATIONS

Kinematic analysis is essential for every serial manipulator study. Subchapter 2.2 presented an Arduino code that allows each motor to go desired angular position. But the nature of serial manipulators focuses one positional purpose. Generally plan is move end effector to desired point. Inverse kinematics is a way to determine deciding how to reach that point. Nevertheless, for repeating and predetermined tasks performing with only forward kinematics, is common due to complexity of inverse kinematics.

3.1 Kinematics of Fanuc S-900 Robot

Figure 3.1 Fanuc S-900 (Robots, 2021)

This chapter will present an example solution of Fanuc S-900 model robot from Tsai (1999). This is articulated, serial, 6 degree of freedom robot just like TQ MA

2000, so solution of TQ MA 2000 will be familiar. Figure 3.1 shows real life model of this robot and Figure 3.1 gives axis representation that will be used in calculations.

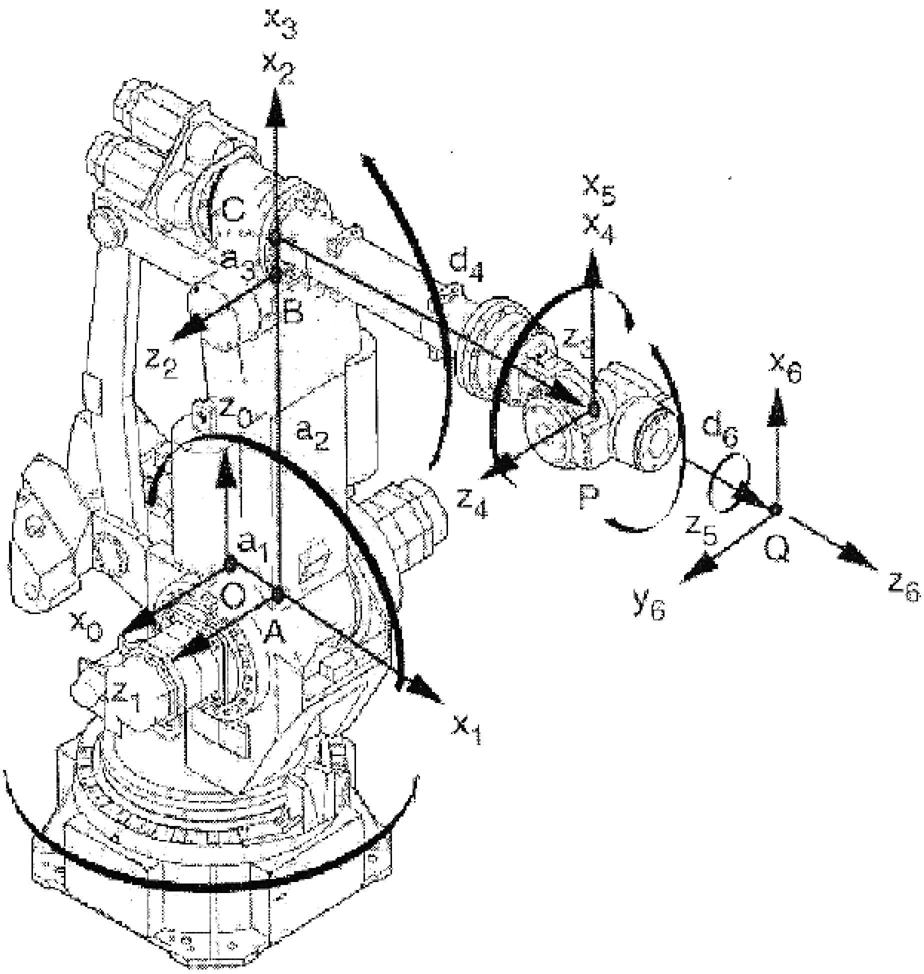

Figure 3.2 Fanuc S-900 axis representation from Tsai (1999)

Tsai (1999) starts with determining Denavit-Hartenberg (DH) parameters of the system. DH parameters are foundation of this kinematic analysis and are given in table below.

Table 3.1 D-H parameters of Fanuc S-900

Joint (i)	α_i	a_i	d_i	θ_i
1	$\pi/2$	a_1	0	θ_1
2	0	a_2	0	θ_2
3	$\pi/2$	a_3	0	θ_3
4	$-\pi/2$	0	d_4	θ_4
5	$\pi/2$	0	0	θ_5
6	0	0	d_6	θ_6

Determining D-H parameters leads to determining D-H transformation matrices. From now on, Cθ statement means that cosine of angle θ and Sθ statement means that sine of the same angle. With these representation method matrices from (3.1) to (3.6) can be created by using D-H parameters.

$$ {}^{0}A_1 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & a_1C\theta_1 \\ S\theta_1 & 0 & -C\theta_1 & a_1S\theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

$$ {}^{1}A_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & S\theta_1 & a_2C\theta_2 \\ S\theta_2 & C\theta_2 & -C\theta_1 & a_2S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2} $$

$$ {}^{2}A_3 = \begin{bmatrix} C\theta_3 & 0 & S\theta_3 & a_3C\theta_3 \\ 0 & 0 & -C\theta_3 & a_3S\theta_3 \\ S\theta_3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3} $$

$$ {}^{3}A_4 = \begin{bmatrix} C\theta_4 & 0 & -S\theta_4 & 0 \\ S\theta_4 & 0 & C\theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4} $$

$$^{4}A_5 = \begin{bmatrix} C\theta_5 & 0 & S\theta_5 & 0 \\ S\theta_5 & 0 & -C\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$^{5}A_6 = \begin{bmatrix} C\theta_6 & -S\theta_6 & 0 & 0 \\ S\theta_6 & C\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Parametric transformation matrix at matrix (3.7) should be equated with multiplication of $^{0}A_3$ and $^{3}A_6$ in that order.

$$^{0}A_6 = \begin{bmatrix} u_x & v_x & w_x & q_x \\ u_y & v_y & w_y & q_y \\ u_z & v_z & w_z & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

This result gives following set of equations.

$$u_x = c\theta_1[c\theta_{23}(c\theta_4 c\theta_5 c\theta_6 - s\theta_4 s\theta_6) - s\theta_{23} s\theta_5 c\theta_6] + s\theta_1(s\theta_4 c\theta_5 c\theta_6 + c\theta_4 s\theta_6)$$
(3.8)

$$u_y = s\theta_1[c\theta_{23}(c\,\partial_4 c\theta_5 c\theta_6 - s\theta_4 s\theta_6) - s\theta_{23} s\theta_5 c\theta_6] - c\theta_1(s\theta_4 c\theta_5 c\theta_6 + c\theta_4 s\theta_6)$$
(3.9)

$$u_z = s\theta_{23}(c\theta_4 c\theta_5 c\theta_6 - s\theta_4 s\theta_6) + c\theta_{23} s\theta_5 c\theta_6 \quad (3.10)$$

$$v_x = c\theta_1[-c\theta_{23}(c\theta_4 c\theta_5 s\theta_6 + s\theta_4 c\theta_6) + s\theta_{23} s\theta_5 s\theta_6] + s\theta_1(-s\theta_4 c\theta_5 s\theta_6 + c\theta_4 c\theta_6)$$
(3.11)

$$v_y = s\theta_1[-c\theta_{23}(c\theta_4 c\theta_5 s\theta_6 + s\theta_4 c\theta_6) + s\theta_{23} s\theta_5 s\theta_6] - c\theta_1(-s\theta_4 c\theta_5 s\theta_6 + c\theta_4 c\theta_6)$$
(3.12)

$$v_z = -s\theta_{23}(c\theta_4 c\theta_5 s\theta_6 + s\theta_4 c\theta_6) - c\theta_{23} s\theta_5 s\theta_6 \quad (3.13)$$

$$w_x = c\theta_1(c\theta_{23} c\theta_4 s\theta_5 + s\theta_{23} c\theta_5) + s\theta_1 s\theta_4 s\theta_5 \quad (3.14)$$

$$w_y = s\theta_1(c\theta_{23}c\theta_4 s\theta_5 + s\theta_{23}c\theta_5) - c\theta_1 s\theta_4 s\theta_5 \quad (3.15)$$

$$w_z = s\theta_{23}\mathrm{c}\theta_4 s\theta_5 - \mathrm{c}\theta_{23}\mathrm{c}\theta_5 \quad (3.16)$$

$$q_x = c\theta_1[a_1 + a_2 c\theta_2 + a_3 c\theta_{23} + d_4 s\theta_{23} + d_6(c\theta_{23}c\theta_4 s\theta_5 + s\theta_{23}c\theta_5)] + d_6 s\theta_1 s\theta_4 s\theta_5$$

(3.17)

$$q_y = s\theta_1[a_1 + a_2 c\theta_2 + a_3 c\theta_{23} + d_4 s\theta_{23} + d_6(c\theta_{23}c\theta_4 s\theta_5 + s\theta_{23}c\theta_5)] - d_6 c\theta_1 s\theta_4 s\theta_5$$

(3.18)

$$q_z = a_2 s\theta_2 + a_3 \mathrm{s}\theta_{23} - d_4 \mathrm{c}\theta_{23} + d_6(\mathrm{s}\theta_{23}\mathrm{c}\theta_4 s\theta_5 - \mathrm{c}\theta_{23}\mathrm{c}\theta_5) \quad (3.19)$$

Due to high amounts of unknown parameters, equations from (3.7) to (3.18) cannot be solved. Therefore, it should be simplified with known or accepted constraints. Intersection of last three axes causes rotations of last three axes' having no effect on the end effector what soever.

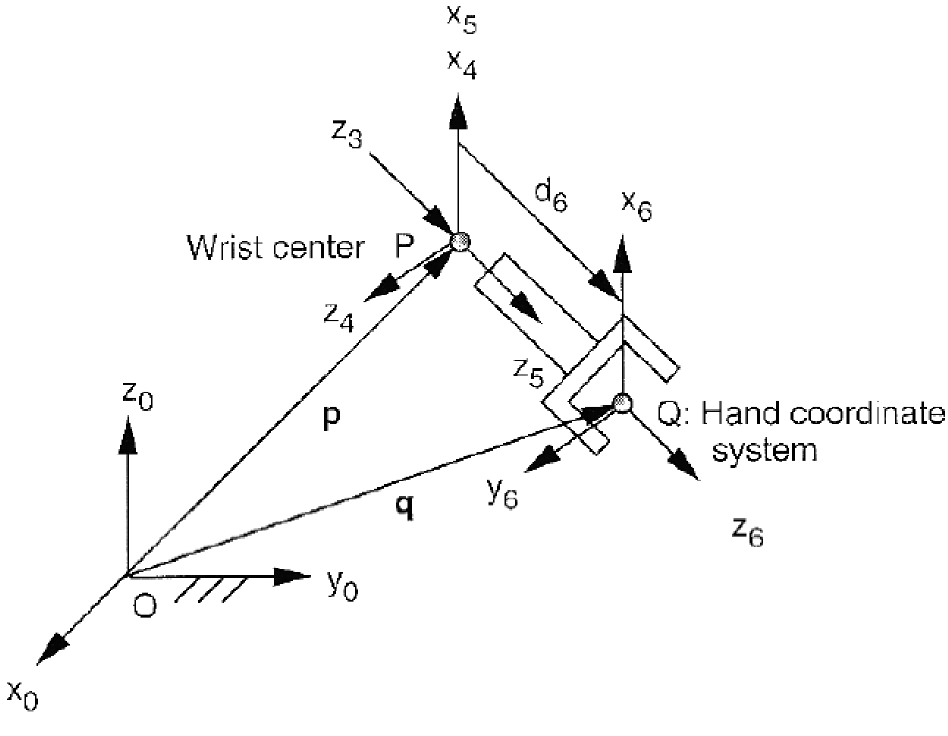

Figure 3.3 Hand coordinate system for center of wrist position from Tsai (1999)

Since last 3 motor has ne effect, first three motors translation matrix can be processed with a vector p, which is a vector from base to fourth motor axis.

$$({}^{0}A_1)^{-1 0}\mathbf{p} = {}^{1}A_3 \, {}^{3}\text{p} \tag{3.20}$$

Solution of this equation system gives following results.

$$\theta_1 = \tan^{-1} \frac{p_y}{p_x} \tag{3.21}$$

Tangent function has two solutions. So, θ_1 has two solutions. These solutions are in 180 degrees opposite direction. Physical constraints of the system has an important role in this decision.

Where

$$\kappa_1 = 2a_2 d_4 \tag{3.22}$$

$$\kappa_2 = 2a_2 a_3 \tag{3.23}$$

$$\kappa_3 = p_x^2 + p_y^2 + p_z^2 - 2p_x a_1 \text{c}\theta_1 - 2p, a_1 \text{s}\theta_1 + a_1^2 - a_2^2 - a_3^2 - d_4^2 \tag{3.24}$$

Known,

$$\frac{\theta_3}{2} = \tan^{-1} \frac{\kappa_1 \pm \sqrt{\kappa_1^2 + \kappa_2^2 - \kappa_3^2}}{\kappa_3 | \kappa_2} \tag{3.25}$$

$$\theta_2 = \text{Atan } 2(s\theta_2, \text{c}\theta_2) \tag{3.26}$$

Until this part, first three positions, hence their transformation matrices were found. Transformation matrix from third to sixth position is unknown. It can be found by consecutive multiplication of transpose of transformation matrix of base to third position and parametric matrix (3.6). When this process is applied and resulting equation system solved following results of other three positions can be found.

$$\theta_5 = \cos^{-1} r_{33} \tag{3.27}$$

$$\theta_4 = \text{Atan } 2(\text{s}\theta_4, \text{c}\theta_4) \tag{3.28}$$

$$\theta_6 = \text{Atan } 2(\text{s}\theta_6, \text{c}\theta_6) \tag{3.29}$$

And this is the methodology of solving kinematic equations of 6 degree of freedom robot arm. Double solutions, that tangent function gave, can be decided with system constraints and desired reaching direction.

3.2 Kinematics of TQ MA 2000

TQ MA 2000 is also a 6 degree of freedom manipulator. With few system specific alterations same methodology can be applied. DH parameters for axis representation that were shown in Figure 2.15, are given in table 3.2.

Table 3.2 D-H parameters of TQ MA 2000

Joint (i)	α_i	a_i	d_i	θ_i
1	$\pi/2$	0	d_1	θ_1
2	0	a_2	0	θ_2
3	0	a_3	0	θ_3
4	$\pi/2$	a_4	0	θ_4
5	$\pi/2$	0	0	θ_5
6	$\pi/2$	0	d_6	θ_6

Based on these parameters following transformation matrices were defined.

$$
{}^{0}A_1 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.30}
$$

$$
{}^{1}A_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2S\theta_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.31}
$$

$$
{}^{2}A_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & a_3C\theta_3 \\ S\theta_3 & C\theta_3 & 0 & a_3S\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.32}
$$

$$
{}^{3}A_4 = \begin{bmatrix} C\theta_4 & 0 & S\theta_4 & a_4C\theta_4 \\ S\theta_4 & 0 & -C\theta_4 & a_4S\theta_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.33}
$$

$$
{}^{4}A_5 = \begin{bmatrix} C\theta_5 & 0 & S\theta_5 & 0 \\ S\theta_5 & 0 & -C\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.34}
$$

$$
{}^{5}A_6 = \begin{bmatrix} C\theta_6 & -S\theta_6 & 0 & 0 \\ S\theta_6 & C\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.35}
$$

In previous chapter matrices from (3.1) to (3.6) were equated to parametric matrix (3.7) and the equation system which is outcome of this operation, was solved equations from (3.8) to (3.20). Same methodology were applied with transformation matrices from (3.30) to (3.35). Solutions are given below.

$$
\theta_1 = a\tan\ 2\left((p_y - d_6a_y), (p_x - d_6a_x)\right) \tag{3.36}
$$

$$\theta_5 = atan\ 2(S_5, C_5) \tag{3.37}$$

$$\theta_6 = atan\ 2\left((o_x S_1 - o_y C_1), (n_y C_1 - n_x S_1)\right) \tag{3.38}$$

$$\theta_2 = atan\ 2(P_2, P_1) + atan\ 2\left(\sqrt{P_1^2 + P_2^2 - N}, N\right) \tag{3.39}$$

$$\theta_3 = \theta_{23} - \theta_2 \tag{3.40}$$

$$\theta_4 = \theta_{234} - \theta_{23} \tag{3.41}$$

3.3 Application

Position scenario that calculated at subchapter 3.2, were applied to TQ MA 2000 model and a whiteboard marker was attached to its end effector. A paper placed at the target that supposed to be perpendicular to intended marker direction.

Figure 3.4 Result of two consecutive actions (Personal archive, 2020)

By this way paper were drawn a dot by a robot. This was done two times per a paper. Then distance between two dots were measured like shown in Figure 3.3 and this measurement procedure was repeated in total 20 times to see the robustness of the system. Results were given in Table 3.3.

Table 3.3 Repeatability Test

Attempt No	Measured Distance (mm)
1	21.35
2	18.23
3	14.2
4	20.25
5	19.74
6	16.86
7	17.27
8	22.52
9	11.95
10	16.25
11	18.67
12	18.24
13	16.12
14	21.23
15	19.97
16	18.68
17	22.19
18	15.83
19	19.45
20	20.33

As seen on Table 3.3. system is not very robust. A system with this much uncertainty can not give reliable sensitivity results. Therefore, sensitivity was not investigated.

3.4 Motion

After results of chapter 3.3, Solidworks CAD model of TQ MA 2000 were used for further testing. To present an example, end effectors movement of 150mm x 150mm rectangular trajectory motion were studied and demonstrated with graphs.

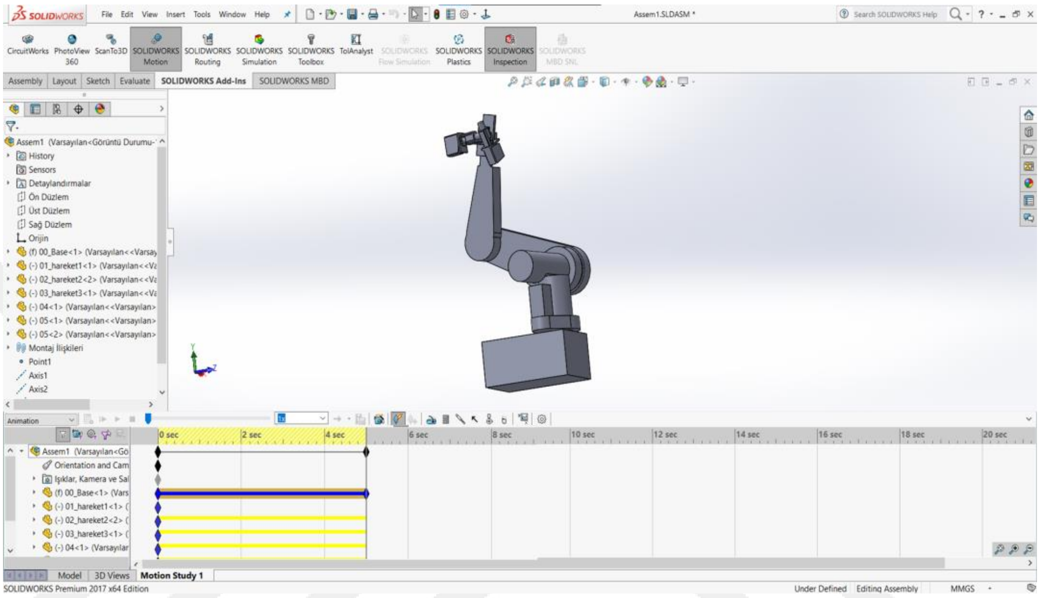

Figure 3.5 Solidworks motion analysis screen

Angular positions of motors from base to end effector referred as teta1, teta2, teta3, teta4, teta5 and teta6 in that order.

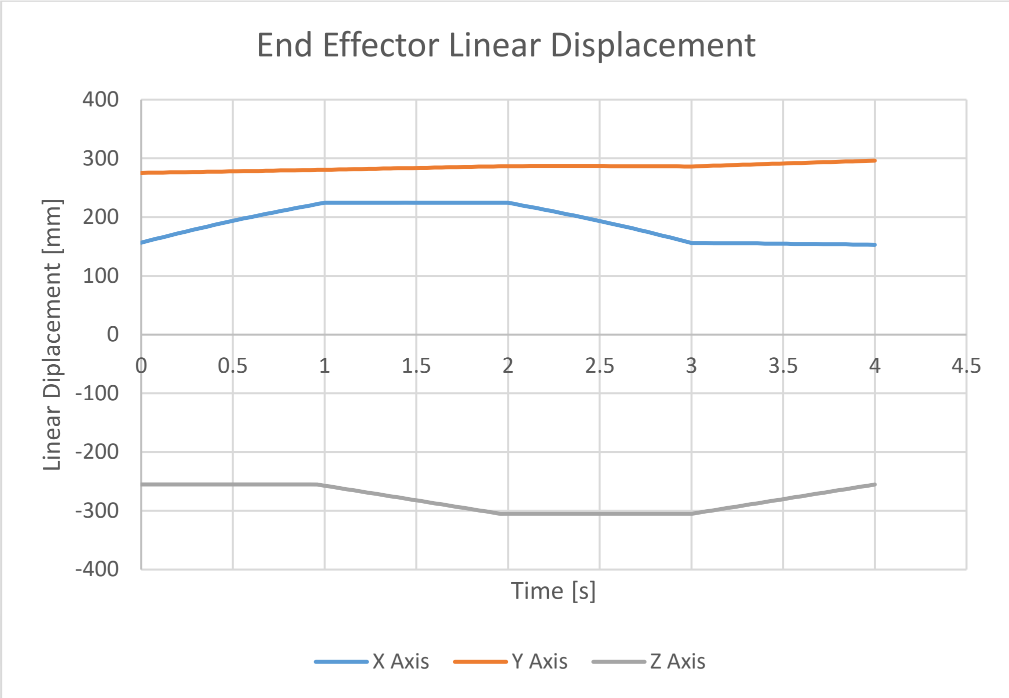

Figure 3.6 End effector linear displacement

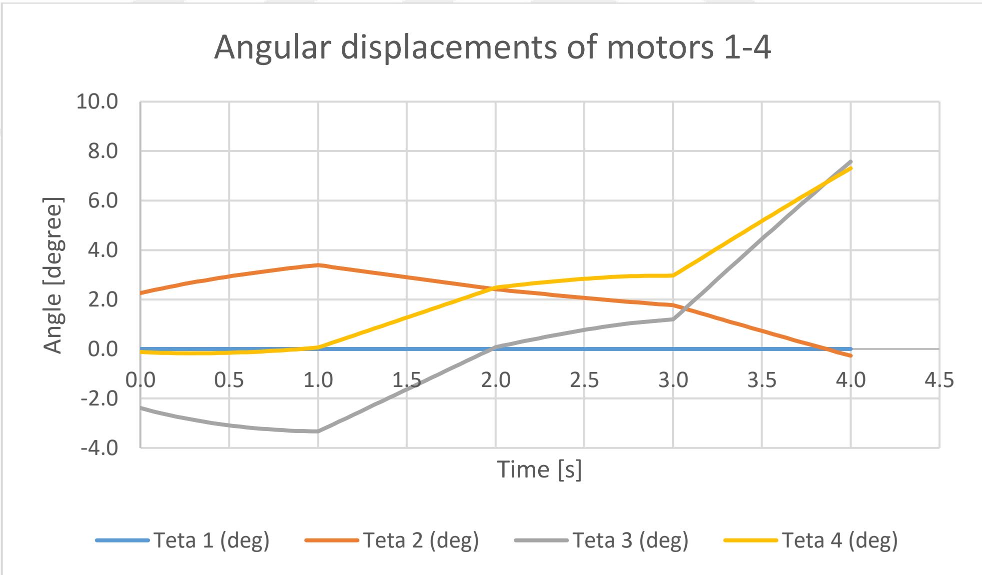

Figure 3.7 Angular displacements of motors 1-4

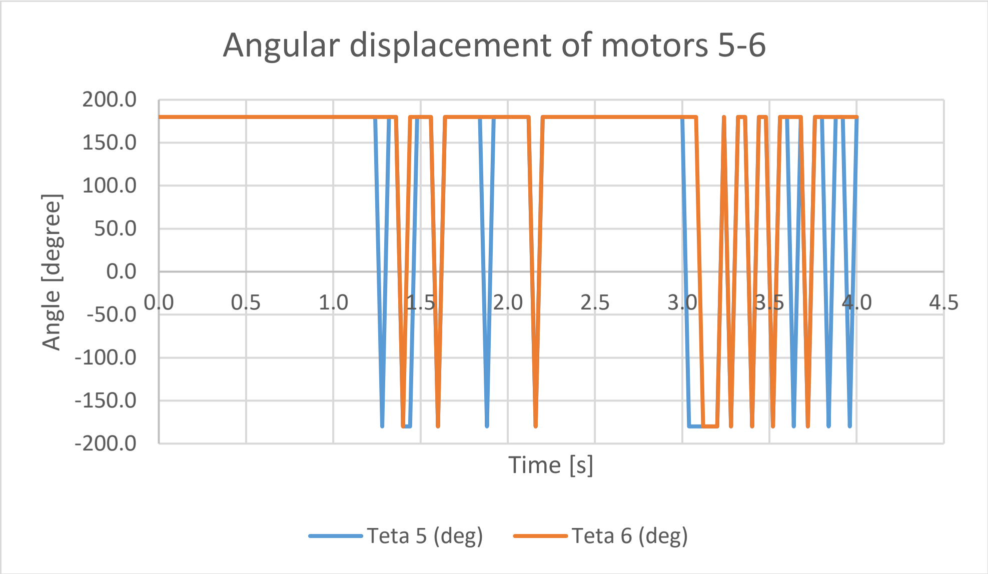

Figure 3.8 Angular displacements of motors 5-6

CHAPTER FOUR
TWO-ARM APPLICATIONS

This chapter will focus on two different types of two hand jobs, twisting and folding. Following 2 subchapters will be about these tasks. This chapter also questions that if there are infrastructural access constraints like pandemic, how much of a two-arm study can be made, by investigating a study without any experimental application. In both two-arm jobs one of the arms considered that has no movement. That arm will not move after arriving desired position and will be referred as the reference arm. The other arm considered to move after the master arm has reached to its destination. This arm will be referred as the support arm. Figure 4.1 presents a schematic illustration of twist and fold scenarios. In this figure, arms represented with color green, joints shown in blue, rotating links according to joint, shown in red. System of connected moving parts A and B, is task to be performed. In each task representation A is the link that will be hold in the same position by the reference arm and B is the link that will be moved to perform the task. Yellow arrows represent movement of the support arm. Since both of the scenarios include performing one arm after the other arm reached to desired position, task can be performed sequentially instead of simultaneously.

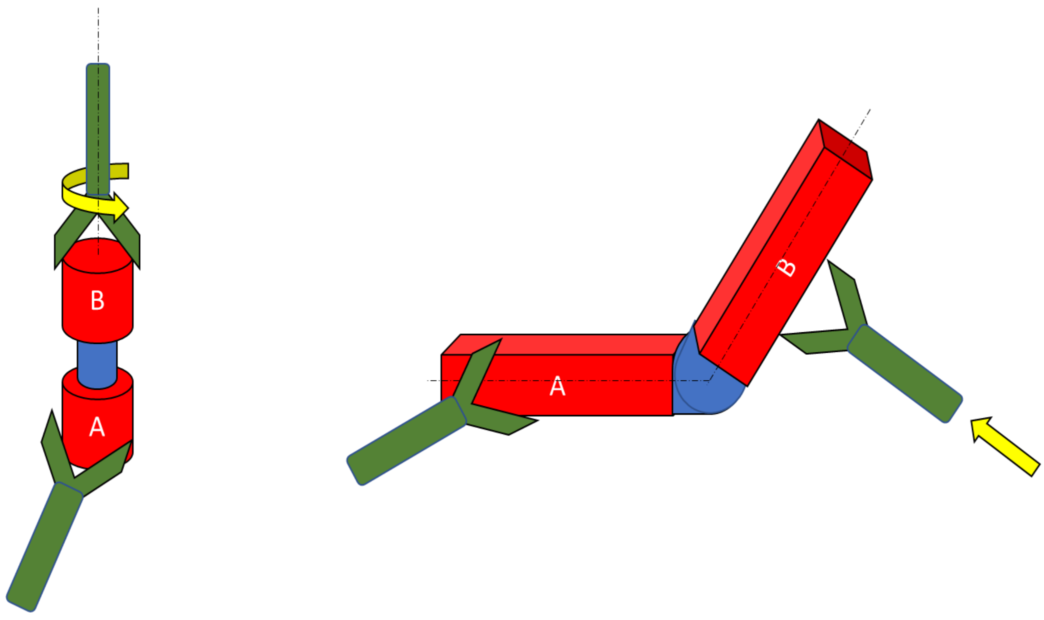

Figure 4.1 Illustration of two hand tasks; twist scenario on the left, fold scenario on the right

4.1 Twist Scenario

Working with two arm comes with more work space limitations. To avoid problems like that, work space was divided by a plane and orientations of the solutions will be chosen accordingly. Work space division was shown in Figure 4.2 and workspace of the reference arm addressed as WS1. For the other arm it addressed as WS2. With this separation, workspace was divided with a plane perpendicular to rotational axis of the twisting.

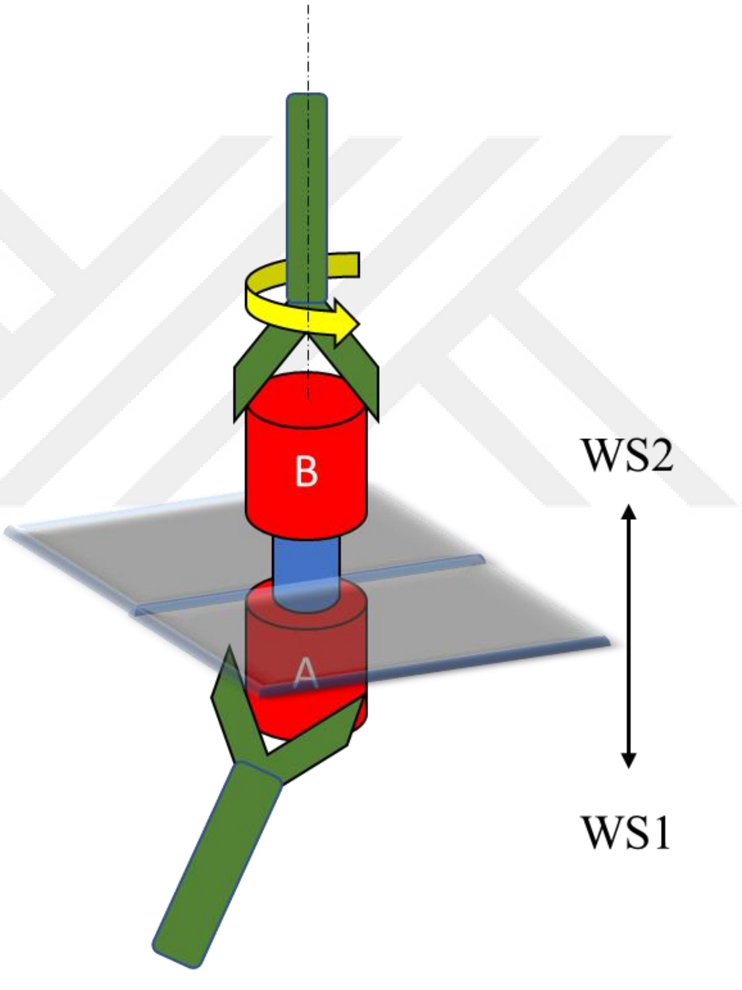

Figure 4.2 Workspace division for a twisting scenario

Figure 2.15 gives the axis representation of base and end effector of TQ MA 2000 as a one arm. But for two arms to cooperate, one must be kinematically defined according to a reference to another. This study mainly focused on robot arm TQ MA 2000 manipulator, hence two arm estimations with the acceptance of both arms being TQ MA 2000. Because of that in Figure 4.3, key points required to make further estimations, can be seen. Where O is the origin of the reference arm and all the other positions defined accordingly and points A and B have coinciding y axis.

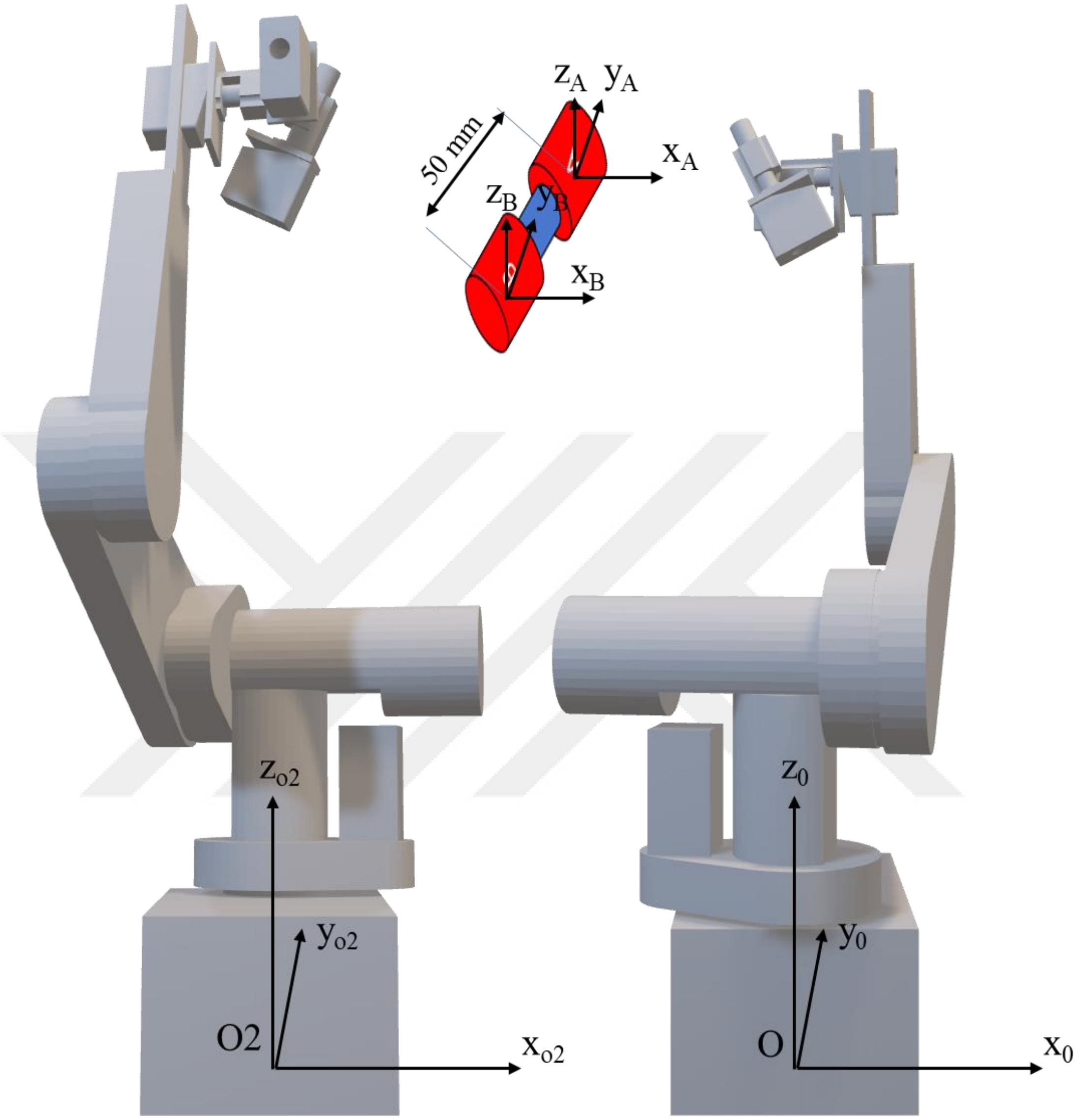

Figure 4.3 Illustration essential points of twisting task case

With respect to the reference frame O, key points were defined as O(0,0,0), A(-25,-10,50), B(-25,-60,50) and O2(-50,-70,0) cm. If these initial conditions applied to solution that presented in subchapter 3.2, solution can be calculated. For this scenario acquired solutions are (-162.6°, -7.9°, 79.8°, 108.1°, 27.7°, 135°) for reference arm and (1.2°, -104.1°, 79.8°, -33.8°, -135°) for the support arm.

4.2 Fold Scenario

As a second case action of folding will be observed. For a better solution workspace division should be made similar to previous subchapter. That was presented in Figure 4.4.

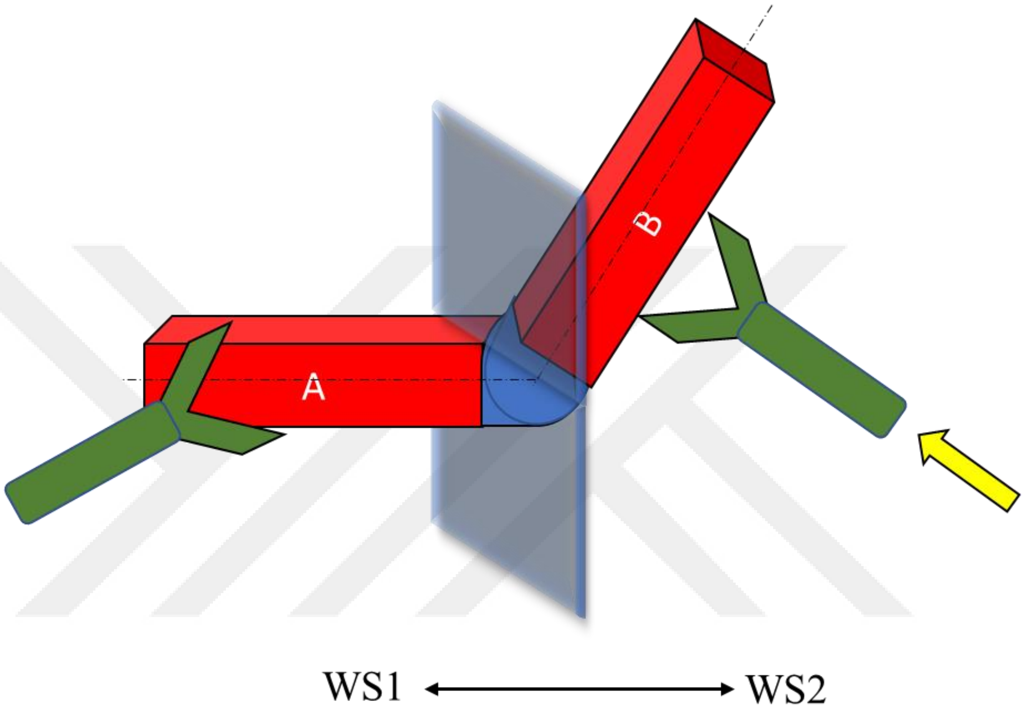

Figure 4.4 Workspace division for a folding scenario

Figure 4.5 shows similar positions to the twisting scenario. The only difference is task and that changes orientation of robots approach, so it leads to a different solution. Initial inputs are same as subchapter 4.1, but for folding task to be accomplished, both arms should approach from below.

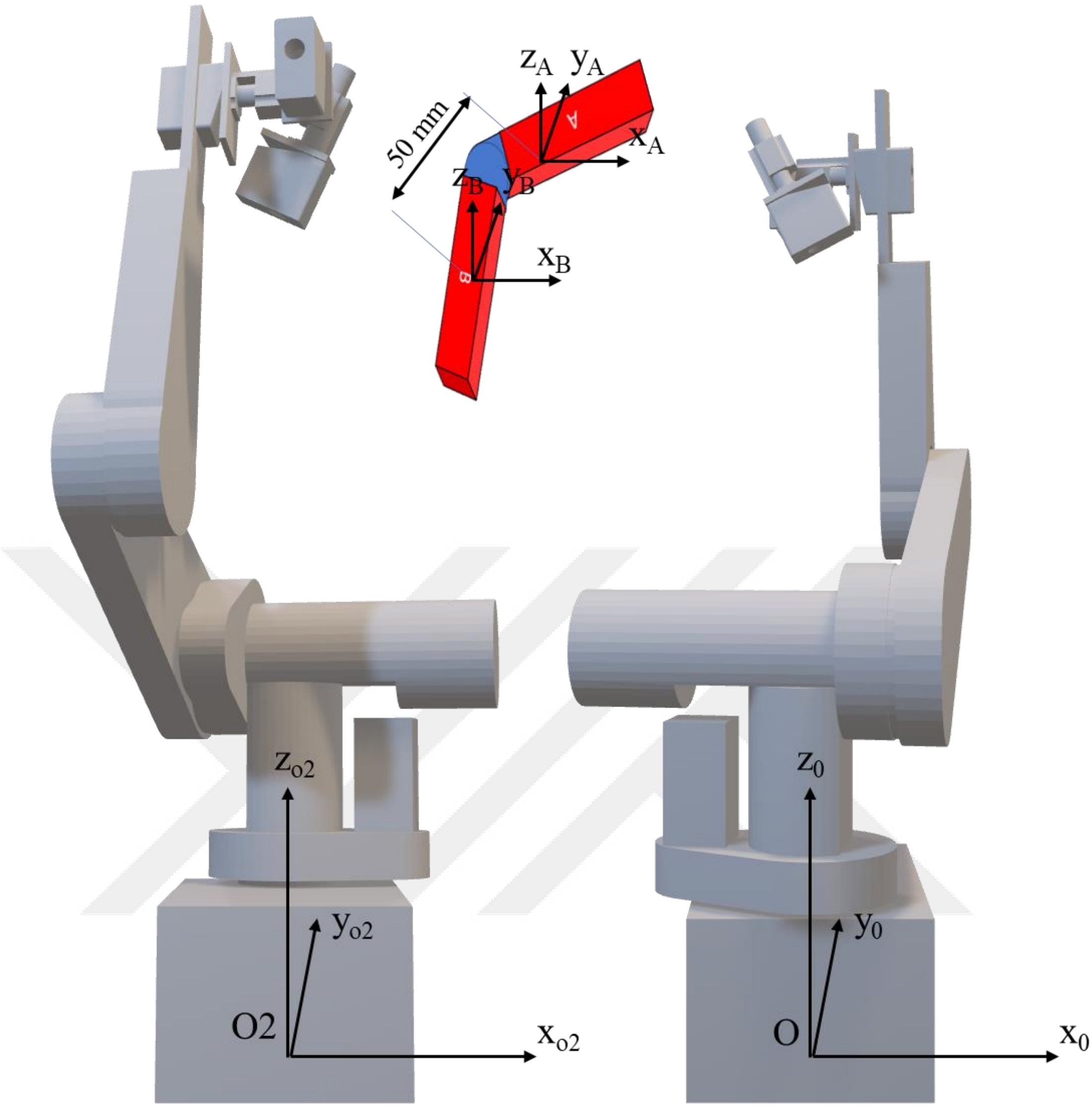

Figure 4.5 Illustration essential points of folding task case

Although this case has almost same initial inputs with twisting case, different solutions may be found. As mentioned before, arms should approach from below and the significant importance for this to be valid, is summation of second and third angles. Because they have parallel rotational axes, hence their total directly effects this orientation. Of course, due to symmetric positions of the arms due to folding task element, negative angles of support arm should be taken. According to these considerations if solution that presented in subchapter 3.2 is applied, a symmetric solution sets of (-162.6°, 75.9°, -79.8°, -176.1°, 27.6°, 135°) for reference arm and

(162.6^{O}, -75.9^{O}, 79.8^{O}, 176.1^{O} ,-27.6^{O} ,-135^{O}) for support arm can be found. Which makes sense that folding action has two opposite directions.

CHAPTER FIVE
CONCLUSIONS

This study has presented renovation endeavors of TQ MA 2000 robot arm, which presents a very little information. Because of that, this study also gives an example of approach to unknown manipulator system. After renovation and identification, experimental setup was presented. Robustness was tested with this experimental setup and an example of kinematic analysis and this was also illustrated. Also, comparatively kinematic analysis was given and extrapolated for an estimation of two dual arm scenarios of twisting and folding. Conclusions were given below;

- Robotic arm studies that necessarily does not require high precision can partially be used for educational purposes and some academic studies. Chapter 4 showed that robustness is not very good for given TQ MA 2000 system. Process of marking the same position on a paper with a pen could not be done consecutively. Average of distance errors found 18.47 mm. therefore, using TQ MA 2000 for any industrial purpose is not a good option. Nevertheless, for educational purposes and development studies on such matters like robustness or sensitivity, it is an adequate option. Study about adapting TQ MA 2000 can be made.
- Foundation of the experimental setup relies on some relatively cheap equipment like dc motor and potentiometer. Because of that, affordability issue for students, researchers and enthusiasts, can be a lesser problem.
- Chapter 4 has a study of estimation for two arm specific scenarios like twisting (bottle opening). Like a pandemic, there may be several infrastructural access limitations. Based on known one arm data, two arm scenarios can be estimated without accessing experimental setup despite the limitations.
- Two arm studies showed that different inverse kinematic solutions can be acquired with same initial parameters depending on the task. Although end effector positions of the twisting and folding scenarios, are the same, because of the nature of the folding scenario, target object should be reached from

below. Solution set were chosen accordingly. This study raises the question of whether a universal task oriented decision making algorithm can be made.

- Software of the application presents some kind of new simplified language based on Arduino’s compiler. This flexible software can be adjusted to different kinds of actuator and sensors easily by just defining connection characteristics. Because of that this study presents a potential universal manipulator software for robot manipulator systems.

REFERENCES

ABB, (2021). *Industrial Robots*. Retrieved February 10, 2021, from https://new.abb.com/products/robotics/industrial-robots.

Al-Faiz, M. Z., & Saleh, M. S. (2011). Inverse kinematics analysis for manipulator robot with wrist offset based on the closed-form algorithm. *International Journal of Intelligent Systems and Applications in Robotics (IJRA), 2* (4), 256-264.

Al-Saedi, F. A. T., & Mohammed, A. H. (2012). Design and implementation of PSO-PID controller for MA2000 robotic manipulator. *IJSET, 2* (9), 1427-1433.

Amazon, (2021). *Elcon India Potentiometers Rotary Master Vol 10k 47k 100k with PCB (10 K Single*. Retrieved February 12, 2021, from https://www.amazon.in/Elcon-India-Potentiometers-Rotary-Master/dp/B07NV2ZHF4.

Beebcontrol, (2019). *MA2000*. Retrieved March 14, 2019, from http://www.riscy.uk/beebcontrol/arms/ma2000/index.html

Buhl, J. F., Grønhøj, R., Jørgensen, J. K., Mateus, G., Pinto, D., & Sørensen, J. K., et al. (2019). A dual-arm collaborative robot system for the smart factories of the future. *Procedia Manufacturing, 38*, 333-340.

Cruz, G. L., Alazki, H., & Hern´andez, R. G. (2018). Super twisting control for Thermo's Catalyst-5 robotic arm. *IFAC-PapersOnLine,51* (13), 303-308.

Hepsiburada, (2021). Retrieved February 12, 2021, from https://productimages.hepsiburada.net/s/3/375/9592703156274.jpg

Hayawi, M. J. (2011). *The closed form solution of the inverse kinematics of a 6-DOF robot*. Retrieved December 7, 2017, from https://www.researchgate.net/publication/239460298_Closed_form_solution_for_the_inverse_kinematics_of_a_redundant_robot_arm

Ideasoft, (2021). Retrieved February 12, 2021, from https://st3.myideasoft.com/idea/cd/40/myassets/products/170/pololu-6v-625rpm.jpg?revision=1547027111

Krüger, J., Schreck, G., & Surdilovic, D. (2011). Dual arm robot for flexible and cooperative assembly. *CIRP Annals, 60* (1), 5-8.

Kuka, (2021). *Industrial robots from KUKA*. Retrieved February 12, 2021, from https://www.kuka.com/en-de/products/robot-systems/industrial-robots

Liu, H., & Dai, J. (2003). An approach to carton-folding trajectory planning using dual robotic fingers. *Robotics and Autonomous Systems, 42* (1), 47-63.

Liu, X., Qiu, C., Zeng, Q., & Li, A. (2015). Kinematics analysis and trajectory planning of collaborative welding robot with multiple manipulators. *Procedia CIRP,81*, 1034-1039.

Mac Tools, (2021). *0" - 6" Digital Caliper with Fractions*. Retrieved February 12, 2021, from https://www.mactools.com/en-us/Specialty-Tools/Measuring-Tools/DDC61232F/0-6-Digital-Caliper-with-Fractions

Manzoor, S., Islam, R. U., Aayman, K., Samad, A., & Iqbal, J. (2014). An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation. *Robotics and Computer-Integrated Manufacturing, 30* (3), 351-362.

Ospennikova, E., Ershov, M., & Iljin, I. (2015). Educational robotics as an inovative educational technology. *Procedia - Social and Behavioral Sciences*, *214* (5), 18-26.

Paunski, Y. K., & Angelov, G. T. (2019). Performance and power consumption analysis of low-cost single board computers in educational robotics. *IFAC-PapersOnLine, 52* (25), 424-428.

Phuong, T. T., Savchenko, M., & Hagiwara, I. (2018). Finite element simulation of robotic origami folding. *Simulation Modelling Practice and Theory, 84*, 251-267.

Robotkits, (2021). *Arduino Mega 2560 R3 Original Made in Italy with box*. Retrieved February 12, 2021, from https://robokits.co.in/arduino/arduino-boards/arduino-mega-2560-r3-original-made-in-italy-with-box

Robotistan, (2021). *L298N Voltaj Regulatörlü Çift Motor Sürücü Kartı(Kırmızı PCB)*. Retrieved February 12, 2021, from https://www.robotistan.com/l298n-voltaj-regulatorlu-cift-motor-surucu-karti.

Robots, (2021). Retrieved February 12, 2021, from https://www.robots.com/images/robots/Fanuc/S-Series/FANUC_S-900iB_400_cutout.png

RS Components, (2021). Retrieved February 12, 2021, from https://res.cloudinary.com/rsc/image/upload/b_rgb:FFFFFF,c_pad,dpr_1.0,f_auto,h_337,q_auto,w_600/c_pad,h_337,w_600/W8509805-01?pgw=1

Shauri, R. L. A., Saiki, K., Toritani, S., & Nonami, K. (2012). Sensor integration and fusion for autonomous screwing task by dual-manipulator hand robot. *Procedia Engineering, 41*, 1412-1420.

Sparkfun, (2021). *L298 Dual Full-Bridge Driver*. Retrieved February 12, 2021, from https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

Stadtler, W. (1995). *Analytical robotics and mechatronics*. New York: McGraw-Hill.

TecQuipment, (n.d.). *MA2000 robot operators manual*. Nottingham: TecQuipment Ltd.

Toolex, (2021). Retrieved February 12, 2021, from https://www.toolex.com/image/cache/catalog/ESC01015-1024x680.JPG

Tsai, L. (1999). *Robot analysis*. New York: Wiley.

Uicker, J. J., Pennock, G. R., & Shigley, J. E. (2011). *Theory of machines and mechanisms* (4^{th} ed). New York: Oxford.

Vatan, (2021). Retrieved February 12, 2021, from https://www.toolex.com/image/cache/catalog/ESC01015-1024x680.JPG

Wu, Q., Li, M., Qi, X., Hu, Y., Li, B., & Zhang, J. (2019). Coordinated control of a dual-arm robot for surgical instrument sorting tasks. *Robotics and Autonomous Systems, 112*, 1-12.

Xu, J., Fuhai, Z., Yu, W., Yu, W., & Yili, F. (2019). Human-like coordination motion learning for a redundant dual-arm robot. *Robotics and Computer-Integrated Manufacturing, 57,* 379-390.

Yang, S., Wen, H., & Jin, D. (2019). Trajectory planning of dual-arm space robots for target capturing and base manoeuvring. *Acta Astronautica, 164*, 142-151.

APPENDICES

APPENDIX A: ARDUINO SOURCE CODE

```
// GENERAL DECLARATIONS

int M[7][7]; //Defining matrix that contains pin numbers of motor controller unit
int mctype[7]; // Defining array for determining type of each motor controller
// number 1 refers to L298 MCU

int p[7]; // Defining variables for keeping position values
String c; // Defining string parameter for Serial Input
int inpv; // Defining integer variable to extract input string
bool active = false; // Defining a parameter which is false until first command given
int deneme;
int oo=-20;
int refz[7];

void setup() {

  //Defining Output Pins Due To Motor Controlling Units
pinMode(0,OUTPUT);pinMode(1,OUTPUT);pinMode(2,OUTPUT);
pinMode(20,OUTPUT);pinMode(21,OUTPUT);pinMode(13,OUTPUT);

//Declaring all motors are driven by L298 Motor Driver parametricly, so that can be applied to different types of drivers
mctype[1]=1;mctype[2]=1;mctype[3]=1;
mctype[4]=1;mctype[5]=1;mctype[6]=1;

//Starting Serial Communication at 9600 baud
Serial.begin(9600);

l298(1,5,1,2);
l298(2,6,24,22);
l298(3,7,35,32);
l298(4,8,44,45);
l298(5,9,47,44);
l298(6,0,3,4);

//Defining analog reference positions of sensors
refz[1]= 770; refz[2]= 674; refz[3]= 179;
refz[4]= 153; refz[2]= 115; refz[3]= 216;

}

void loop() {

//Input data___________________
```

```
 if (Serial.available() > 0) {     // is a character available?
  c = Serial.readStringUntil('\n');
inpv=c.toInt();
active = true;

}//_____________________________

//Acquiring and Interpreting sensor datas
interpret(1,110.32);
interpret(2,109.32);
interpret(3,238.37);
interpret(4,204.6);
interpret(5,206.67);
interpret(6,202.57);

Serial.println(analogRead(1));

}

void l298(int motorno,int enable, int input1, int input2) { // l298
function that addresses pin numbers of L298 motor controller unit
M[motorno][1]= enable; pinMode(enable,OUTPUT);
M[motorno][2]= input1; pinMode(input1,OUTPUT);
M[motorno][3]= input2; pinMode(input2,OUTPUT);
deneme = enable;
}

void turn(int motorno,  int sped) { //turn function commands related
motor drivers to turn motor bidirectionally
if(mctype[motorno]==1)  //Detection of L298 Motor Driver
{
analogWrite(M[motorno][1],sped);     //overriding  pwm  speed  signal
with a value between -255 and 255
if(sped<0){digitalWrite(M[motorno][2],HIGH);digitalWrite(M[motorno][
3],LOW);} //positive direction settings for positive speed value
if(sped>0){digitalWrite(M[motorno][2],LOW);digitalWrite(M[motorno][3
],HIGH);} //negative direction settings for negative speed value
  }

  if(mctype[motorno]==2)  //Detection of MC33926 Motor Driver
  {}

}

void fms(int motorno){  // Fast motor stop program
if(mctype[motorno]=1){digitalWrite(M[motorno][1],HIGH);digitalWrite(
M[motorno][2],LOW);digitalWrite(M[motorno][3],LOW);}
}

void but1(int motorno, int analog){  // button control 1 - pot
int c;
if(analog>700) {c=100;}
else if(analog<400) {c=-100;}
else {c=0;}
turn(motorno,c);
```

```
}

void go(int motorno, int aci){   // driving commanded motor to desired position
int hiz;
if(p[2]<(aci-1)){hiz=-100;}
if(p[2]>=(aci-1)&&p[2]<=(aci+1)){hiz=0;}
if(p[2]>(aci+1)){hiz=100;}
turn(motorno,hiz);
}

void drive(int motorno, int aci,int hiz){   // driving commanded motor to desired position
if(p[2]<(aci-1)){hiz=-hiz*1;}
if(p[2]>=(aci-1)&&p[2]<=(aci+1)){hiz=hiz*0;}
if(p[2]>(aci+1)){hiz=hiz*1;}
turn(motorno,hiz);
}

void interpret(int potno,int m){   // Interpreting sensor data to position values
p[potno]= (analogRead(potno), - refz[potno])*5*m/1023 ;
}
```