TWO DEGREES OF FREEDOM FORK MOUNT ACTUATING MECHANISM DESIGN FOR OPTICAL TELESCOPE

A MASTER'S THESIS

In

Mechatronics Engineering

Atılım University

By

YAŞAR YILDIRAN

JULY 2015

TWO DEGREES OF FREEDOM FORK MOUNT ACTUATING MECHANISM DESIGN FOR OPTICAL TELESCOPE

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

YAŞAR YILDIRAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF MECHATRONICS ENGINEERING

JULY 2015

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

Prof. Dr. K. Ibrahim AKMAN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Abdulkadir ERDEN

Head of Department

This is to certify that we have read the thesis “Two Degrees Of Freedom Fork Mount Actuating Mechanism Design for Optical Telescope” submitted by Yaşar YILDIRAN and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Sedat NAZLIBİLEK

Supervisor

Examining Committee Members

Assoc. Prof. Dr. Sedat Nazlıbilek _____

Assoc. Prof. Dr. Fuad Aliew _____

Asst. Prof. Dr. Bülent İrfanoğlu _____

Assoc. Prof. Dr. Cahit Yeşilyaprak _____

Assoc. Prof. Dr. Hakan Volkan Şenavcı _____

Date: 15.07.2015

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Yaşar YILDIRAN

Signature:

ABSTRACT

TWO DEGREES OF FREEDOM FORK MOUNT ACTUATING MECHANISM DESIGN FOR OPTICAL TELESCOPE

YILDIRAN, Yaşar

M.S., Mechatronics Engineering Department

Supervisor: Assoc. Prof. Dr. Sedat Nazlıbilek

July 2015, 85 pages

The aim of this thesis is to design two degrees of freedom fork mount mechanism for optical telescopes develop the electronics, mechanics and software and to build a prototype. Telescope mount design and prototype have been done for small class optical telescopes. The mount has been able to steer optical telescopes that weigh up to 35 kilograms. Fork arms distance of the mount permits 20 to 40 cm diameter sized primary mirrors to be used.

The system mechanic requirements were determined and proper design has been done. Actuators for motion and electronic hardware necessary to control the system have been provided. System controls are done with a two layer block model on Simulink. Simulink is able to control actuators via our main controllers Arduino Mega 2560 at external mode. System's closed loop control has been done with well known Proportional-Integral-Derivative (PID) controller.

The completed prototype has been tested in target and tracking modes. Test results are provided in the thesis.

Keywords: Telescope Mount, Alt-Az Telescopes, Telescope Control Structure, Arduino Modeling in MATLAB - Simulink

ÖZ

OPTİK TELESKOPLAR İÇİN İKİ SERBESTLİK DERECELİ ÇATAL KURGU TASARIMI

YILDIRAN, Yaşar

Yüksek Lisans, Mekatronik Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Sedat NAZLIBİLEK

Temmuz 2015, 85 sayfa

Bu tez ile optik teleskoplar için iki serbestlik derecesinde hareket mekanizmasına sahip çatal kollu teleskop montürünün elektronik, mekanik ve yazılımsal tasarımının ve prototipinin yapılması amaçlanmıştır. Teleskop montürü küçük sınıf teleskoplar için tasarlanmış ve prototiplenmiştir. Protatiplenen teleskop montürü; 35 kg ağırlığındaki optik teleskopları yönlendirebilecek kapasitededir. Teleskop montürünün çatal kolları arasındaki mesafe; 20 cm ile 40 cm ayna çapına sahip optik teleskopların takılmasına imkan verecek niteliktedir.

Sistem mekanik gereksinimleri tespit edilmiş buna uygun mekanik tasarım yapılmıştır. Hareketi sağlayacak eyleyiciler ve sistem kontrolü için gerekli elektronik altyapı temin edilmiştir. Sistemin kontrolü Simulink'te oluşturulmuş iki katmanlı blok model ile yapılmaktadır. Simulink harici modda ana kartımız olan Arduino Mega üzerinden eyleyicilerin kontrolüne imkan vermektedir. Sistemin kapalı devre kontrolü iyi bilinen PID kontrolcü ile yapılmıştır.

Yapılan prototipin hedefleme ve takip kabiliyeti test edilmiş, elde edilen test sonuçları değerlendirilmiştir.

Anahtar Kelimeler: Teleskop Kurgusu, Teleskop Montürü, Alt-Az Teleskoplar, Teleskop kontrol mimarisi, MATLAB – Simulink ile Arduino kart kullanımı

To My Wife, Friends and Brothers

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Assoc. Prof. Dr. Sedat NAZLIBİLEK for his guidance and insight throughout the research. Thanks also go to my friend Amir Nobahar Sadeghi and all administrative and academic staff from Department of Mechatronics Engineering. Thanks also go to technicians in the machine shop, especially Mehmet Çakmak and my brothers Tolga and Fatih. To my wife, Başak, I offer sincere thanks for her continuous support and patience during this period.

TABLE OF CONTENTS

ABSTRACT.....iii

ÖZ.....iv

DEDICATION.....v

ACKNOWLEDGMENTS.....vi

TABLE OF CONTENTS….....vii

LIST OF TABLES x

LIST OF FIGURES.....xi

LIST OF ABBREVIATIONS.....xiii

LIST OF NOMENCLATURE xiv

CHAPTER

1. INTRODUCTION.....1

2. BACKGROUND INFORMATION….....3

2.1 Telescope Tube Assembly (OTA).....4

2.1.1 Refracting telescopes.....5

2.1.2 Reflecting telescopes.....5

2.1.3 Catadioptric telescopes.....6

2.2 Telescope Mounts (TM).....6

2.2.1 Current Telescope mounts types.....7

2.2.1.1 Alt-Azimuth.....7

2.2.1.2 Equatorial.....8

3. CONCEPTUAL DESIGN 9

3.1 Design Criteria and requirements 9

3.2 Equipments.....10

3.2.1 Optical Tube Assembly (OTA)….....10

3.2.2 Telescope Mount (TM).....11

3.3 Electro-Mechanical Model Design for Prototype.....11

3.3.1 Mechanical Drawings.....11

3.3.2 DC Motor selection.....11

3.3.2.1 DC Motors.....12

3.3.2.2 Motor Brackets.....12

3.3.3 Worm and worm wheels.....13

3.3.4 Ball bearings.....14

3.3.5 Main body.....14

3.3.5.1 Altitude axis structure.....15

3.3.5.2 Azimuth axis structure.....15

4. CONTROL SYSTEM …….....19

4.1 Control System Design…….....19

4.1.1 Main Controller PC.....20

4.1.2 Control Board.....20

4.1.3 Dual Motor Driver.....21

4.1.4 DC motors.....22

4.1.5 Battery.....23

4.2. Electromechanical system transfer functions and model design.....23

4.3 System model representation with Simulink.....27

5. PROTOTYPE DEVELOPMENT AND CONSTRUCTION.....32

5.1 Production and assembling of mechanic components.....32

5.2 Electronics and Software implementation.....35

5.2.1 Arduino based motor control (electronic).....36

5.2.2 Encoders 36

5.3 Real system (plant) and PID tune..... 38

5.3.1 System identification..... 43

5.3.2 Manual tuning..... 47

6. CONCLUSIONS AND FUTURE WORKS..... 50

6.1 Conclusions..... 50

6.2 Future works..... 51

REFERENCES 53

APPENDICES

A. DATASHEETS OF EQUIPMENTS.....55

B. TELESCOPE MECHANICAL DESIGN.....66

C. ENCODER READ C CODE.....82

LIST OF TABLES

1. Telescope Profile of Turkey.....3

2. Motor dimensions and general specifications 12

3. Mechanical properties for M1 and M2 Masses.....17

4. Altitude and Azimuth Axis Structure 26

5. Constants for models.....27

6. Properties of tf2 Transfer Function 45

LIST OF FIGURES

1. Telescope Marketing..... 2
2. Schematic of Refracting Telescope ……..... 5
3. Light path in a Cassegrain Telescope 6
4. Light path in a Schmidt-Cassegrain ……..... 6
5. Horizontal Coordinate System and Alt-Azimuth mount KAF-TM11 ……..…... 7
6. Equatorial Coordinate System and Equatorial mount sample ……..... 8
7. Optical Tube Assembly ……..... 10
8. Worm and Worm Wheel …..……..... 13
9. Ball bearing's pictures and sections……. 14
10. M1 Altitude Axis Structure .……...…. 15
11. M2 Azimuth Axis Structure ..……..……...…. 16
12. Variable Fork Arms ……..……..……..……...…. 17
13. Telescope mount design's exploded view, Telescope mount design's assembled view……...…….. 18
14. Analog Control System in S-Domain ……...…..……...… 19
15. Sampled-Data Control System in Z-Domain ....……...…..……...… 19
16. Control System …….……..……...… 20
17. Arduino Mega 2560 Controller……..……...…….. 21
18. Dual VNH5019 Motor Driver Shield with an Arduino …...……...……... 22
19. 12V Brushed DC motor, a 131:1 Gearbox and a Quadrature Encoder …….. 22
20. 5200 mAh Battery Pack……... 23
21. The Electrical model of the DC Motor ..…….....……23
22. Equivalent typical mechanical loading on a motor……..……... 24
23. Relationship between the motor, gear and load……... 25
24. Torque – Speed Curve ..……...…….... 26
25. System mathematical model representation…….…..... 27
26. P Controller responses……..……... 28
27. PID Controller responses…….…… 30
28. 12 mm Thickness plate size and manufacturing model ….……..... 32
29. The parts after cutting operations ……...…….... 32
30. At the beginning of the assembling phase ……..…… 33
31. Some of details of mount phase .....…….…….... 33

32. Completed Telescope Mount and Telescope Optics ……..…….. 34
33. Control System Structure ..…….…..…... 35
34. MATLAB Simulink Arduino Package and Arduino Mega 2560……...…… 35
35. Electronic hardware implementation…….....**.** 36
36. Encoder behavior on oscilloscope screen, Count plot and Slit …….....**.** 37
37. Main control page on Simulink…….....**.** 39
38. Arduino control page (plant) on Simulink……..…. 39
39. Close Loop Control (CLC) unity feedback for 20°, 50°,0°,-30° input response. 40
40. Azimuth axis CLC, P Controller for 30°, 10° input response ……...…..… 40
41. Altitude axis CLC, unity feedback for 30°, 0° input response ……**...** 41
42. Azimuth axis CLC, P Controller for 180° input response …..…… 41
43. Azimuth axis CLC, P Controller for 5°, 15°, and 30° input response ……..... 42
44. Azimuth axis CLC, PID Controller for 5° input response……..……. 42
45. Altitude & Azimuth axis P Controller 30°, 60°, 120°, 45° input response …….. 43
46. Matlab System Identification Toolbox and part of procedure ….…….. 44
47. Measured data and estimated multiple models…….……... 44
48. Step Response of Transfer Function 2 (Tf2) …….…….. 45
49. SI Based system control model ...…….……... 46
50. SI Based system control model step response **..**…….…. 46
51. Actual system response for both axes for SI based PID Controller **.**…….**..** 47
52. Actual system response for both axes for Manual tune based PI Controller ….. 48
53. Actual system ramp input response for both axes for PI Controller …….**.** 49
54. Completed telescope mount prototype ……...…… 51

LIST OF ABBREVIATIONS

OTA	-	Optic Tube Assembly
TM	-	Telescope Mount
RA	-	Right Ascension Axis
Dec	-	Declination Axis
Alt	-	Altitude axis
Az	-	Azimuth Axis
ARX444	-	Polynomial Models
CCW	-	Counterclockwise Direction
CLC	-	Close Loop Control
CW	-	Clockwise Direction
DAQ	-	Data Acquisition Card
DC	-	Direct Current
PWM	-	Pulse with Modulation
PID	-	Proportional, Integral, Derivative
SI	-	System Identification
SS1	-	State Space Model Estimation
TF1	-	Transfer Function
VDC	-	Volt Direct Current
CCD	-	Charge-Coupled Device
M1, 2	-	Mirror
PC	-	Personal Computer
VIN	-	Voltage Input
GRN	-	Ground
IC	-	Integrated Circuit
IR	-	Infrared
CPR	-	Counts per Revolution
TCPTR	-	Total Count per Telescope Revolution
QEI	-	Quadrature Encoder Interface

NOMENCLATURE

ω_{tt} - Angular Velocity

$\omega_{no-load}$ - No load Velocity

$v_b(t)$ - Back Electromotive Force

K_b - Back Emf Constant

$\omega_m(t)$ - Angular Velocity of Motor

$e_a(t)$ - Input Voltage

$T_m(s)$ - Motor Torque

J_a - Motor Moment of Inertia

D_a - Motor Viscous Damping Coefficient

J_m - Total Moment of Inertia

D_m - Total Viscous Damping Coefficient

J_L - Moment of Inertia of Load

D_L - Viscous Damping Coefficient of Load

L_a - Armature Inductance

R_a - Armature Resistance

K - Numerator Coefficient

α - Denumerator Coefficient

T_{stall} - Stall Torque

N_1 - Worm Gear tooth

N_2 - Worm Wheel tooth's

K_t/R_a - Stall Torque Voltage Relation coeff.

$tf2$ - Transfer Function 2

CHAPTER 1

INTRODUCTION

Optical telescopes have been used for identifying object's, places, properties and physical conditions for a long time. Optical telescopes used for astronomical purposes may be considered to be composed of two structural parts, namely, Optical Tube Assembly (OTA) and Telescope Mount (TM). In this thesis, a two-degrees of freedom fork mount actuating mechanism has been researched, modeled and a prototype of small class optical telescope has been produced.

Researches show that optical telescopes have been widely used in the following fields and purposes.

They are used in departments of universities such as astronomy, astrophysics, and space research for educational purposes, experimental works, and observations.
They are used in small observatories for scientific observations.

They are used in big observatories as auxiliary equipments. These auxiliary devices can be atmospheric turbulence measurement device (MASS / DIMM) or finderscope.

They are used in related departments of universities or observatories for popular astronomical activities. People take information on space and celestial objects with these activities.

They are used in primary and high school for educational and social activities. People who meet with telescopes in early ages, they may have an experience for determining their locations and they may have direction finding abilities.

They are used by a number of amateur astronomers for producing or doing observations with these telescopes.

They are used for following and observing purposes by unmanned air vehicles (UAV) or artificial satellites.

Market share of this class of optical telescopes is quite big in total (Fig.1). Meade and Celestron are two major manufacturers of telescopes. The telescopes offered by these companies were evolved from good to excellent within years. Meade and Celestron have made a much bigger presence in the "consumer" telescope market in the past few years [1]. Both Meade and Celestron still make some outstanding products. However, nowadays the problem is that it is difficult to find the good ones in consumer outlets in recent years.

Figure1: Telescope Marketing [1]

We compared the telescopes of these companies' to each other with respect to price and functionality. This comparison gives us information about our telescope mount prototype price and specifications. Meade LX200 Telescope model price is 12.000 TL with 8" primary OTA, Alt-Az mount, and tripod and GOTO property. Celestron CPC Deluxe HD telescope model price is 11.000 TL with 11" primary OTA, Alt-Az mount, tripod, finderscope, NexRemote control software and GOTO property. These prices do not include any tax, custom expense and shipping.

CHAPTER 2

BACKGROUND INFORMATION

Research and development activities in this area are made both by researchers who have graduated from the department of astronomy and astrophysics or by those who are working in this field in Turkey. Telescope mirror workshop was organized at the Istanbul Kültür University in 2008. Mirror grinding and polishing have been shown during this workshop. In 2009, Istanbul Kültür University organized another telescope making workshop as well. The operations such as grinding and calibrations of the mirrors, and the final construction of telescope mounts were all accomplished in only five days. The result was 100 telescopes which were expected to serve thousands of people all over Turkey, because the participants, in particular the teachers, have been encouraged to share the telescopes with their schools and surrounding population. The teachers have also agreed to teach telescope making as part of their classroom activities, so that many students can learn how to make their own telescopes [2].

In 2006 www.atmturk.org [3] web site was created to bring amateur telescope makers close together. This web site has shown what the last telescope making activities are and what can we do in Turkey. Two examples that were completed in the workshop are the 15" primary mirror Dobsonian mount telescope and, 8" primary mirror German equatorial mount telescope.

In addition, Turkish defense related companies (Aselsan, Havelsan, etc.) have big potential and knowledge on this subject.

Turkey has several kinds of telescopes for scientific observation purposes. However, these optical telescopes are all foreign origin. Table 1 shows these telescopes. In this Table, observatories, telescope name/code, telescope aperture size, telescope mount types and origin are given.

Table 1: Telescope Profile of Turkey

	Observatory	Telescope name / code	Telescope Aperture Size (cm)	Mount Type	Origin	Explanation
1	TUBITAK National Observatory (TUG)	RTT150	150	Fork Mount	Russia	Turkey-Russia Assoc
2	TUBITAK National Observatory (TUG)	YT40	40	Fork Mount	US	
3	TUBITAK National Observatory (TUG)	T60	60	Fork Mount	US	
4	TUBITAK National Observatory (TUG)	T100	100	Fork Mount	US	
5	TUBITAK National Observatory (TUG)	ROTSEIIId	40	Fork Mount	US	ROTSE

6	Ankara University Kraiken Obs. (AUKR)	Kreiken	40	Fork Mount	US	
7	Ankara University Kraiken Obs. (AUKR)	Maksutov	30	Fork Mount	Sweden	
8	Istanbul University Observatory	Astrograph	30	Eng. Mount	England	
9	Istanbul University Observatory	İST60	60	German Equ.	Germany	Çanakkale
10	Ege University Observatory	A48	48	Fork Mount	Germany	
11	Ege University Observatory	T40	40	Fork Mount	Germany	
12	Ege University Observatory	T35	35	Fork Mount	Germany	
13	Ege University Observatory	T30	30	Fork Mount	Germany	
14	Çanakkale University Observatory	T122	122	Alt-Az Mount	Germany	
15	Çanakkale University Observatory	T40	40	Fork Mount	US	
16	Çanakkale University Observatory	T30a	30	Fork Mount	US	
17	Çanakkale University Observatory	T30b	30	Fork Mount	US	
18	**Atatürk University Observatory (DAG)**	**ATA400**	**420+**	**Alt-Az Mount**	**-**	**Under const.**
19	Atatürk University Observatory (DAG)	ATA50	50	German Equ.	Austria	test phase
20	Atatürk University Observatory (DAG)	DIMM30	30	Fork Mount	US	Under const.

As seen in Table 1, Turkey cannot compete with other countries according to aperture sizes. As you notice that the telescope described in line 18 of Table 1, which is a DAG's 400 plus class telescope, will be the biggest initiative on this field in Turkey. This investment will hopefully improve the ability of Turkey in astronomy and astrophysics field.

Maintenance, technical support, repair, development, modernization, and even production of these telescopes are subject of engineering disciplines. All these requirements have been welcomed by astronomers or astrophysicists. Turkey's engineering departments have concerns with related subjects as solar tracking projects, satellite tracking projects, space explorations for last decade. This was the most motivating situation for the preparation of this thesis. We believe that this research will reinforce the communication between engineering departments with fundamental sciences departments. We hope that home production will be excited with this thesis on this field.

As mentioned before, a telescope has two structural parts. Optical Tube Assembly (OTA) is the payload for the Telescope Mount (TM). Telescope mount is designed according to the physical properties or intended use of the OTA. For this reason, first telescope optics will be considered in the following section.

2.1 Telescope Tube Assembly (OTA):

Optical telescopes can be classified by three primary optical designs which are refractor, reflector, and catadioptric. They all have their different advantages and

disadvantages and they are used in different areas of professional and amateur astronomy [4].

2.1.1 Refracting telescopes

Refracting telescopes are the most common form of the telescope. They have a long, thin tube where light passes in a straight line from the front objective lens directly to the eyepiece at the opposite end of the tube as seen in Fig.2 [5]. Some of the advantages are that they are easy to use, simple design and no need for maintenance. They are excellent for lunar, planetary and binary stargazing, especially with larger apertures. Some of the disadvantages are that they are heavier, longer and bulkier than equivalent aperture reflectors and catadioptric. Also they are more expensive than the reflectors having the same apertures.

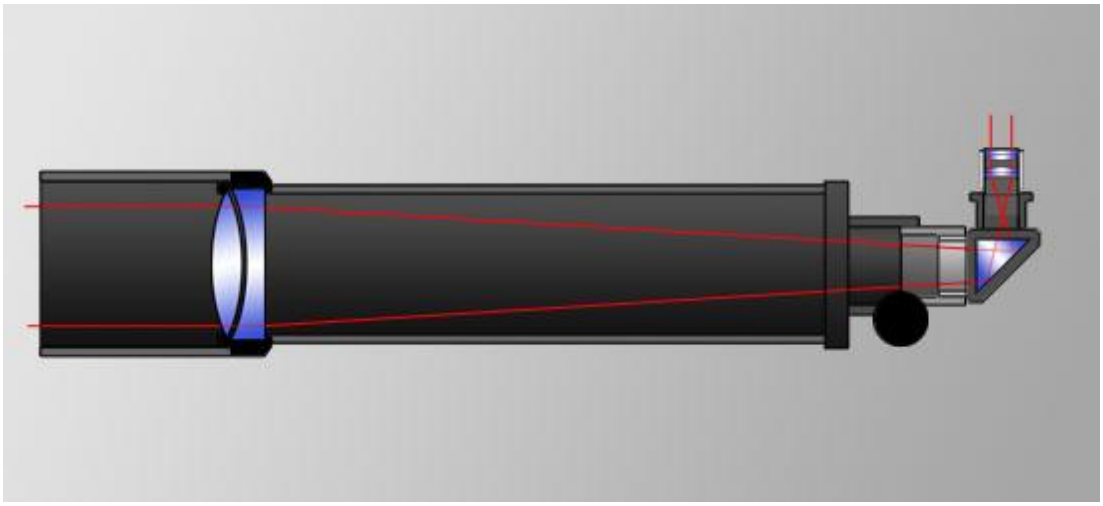


Figure 2: Schematic of Refracting Telescope [5]

2.1.2 Reflecting telescopes

Reflecting telescopes use a huge concave parabolic mirror instead of a lens to gather and focus the light to a flat secondary mirror that in turn reflects the image out of an opening side of the main tube (Fig.3). One can look through an eyepiece on the side of the tube up near the top. Some of the advantages are that they are easy to use and even construct, reasonably compact and portable, low in optical irregularities and deliver very bright images [5].

They are excellent for faint deep sky objects such as distant galaxies, nebulae and star clusters due to their larger apertures for light gathering. A reflector has the least costs per inch of aperture, when compared to refractors and catadioptrics, since mirrors can be produced at less cost than lenses. Some of the disadvantages are that they are not suited for terrestrial applications; slight light loss due to secondary

obstruction when compared with refractors. Reflectors may require a little more care and maintenance.

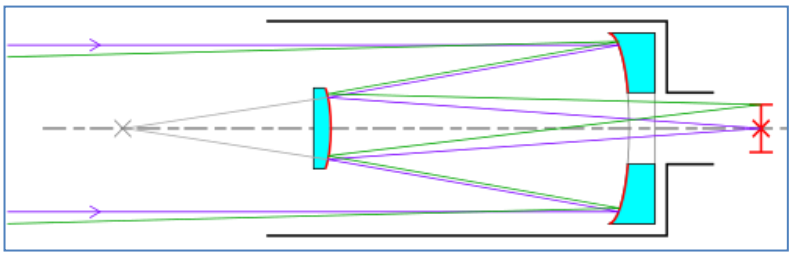


Figure 3: Light path in a Cassegrain Telescope [5]

2.1.3 Catadioptric telescopes

Catadioptric telescopes use a combination of mirrors and lenses to fold the optics and form an image. Catadioptrics are the most popular type of instrument, with the most modern design and larger apertures. There are two popular designs, the Schmidt-Cassegrain (Fig.4) and the Maksutov-Cassegrain. In the Schmidt-Cassegrain, light enters through a thin aspheric Schmidt correcting lens, then strikes the spherical primary mirror and is reflected back up the tube to be intercepted by a small secondary mirror. The mirror then reflects the light out an opening in the rear of the instrument where the image is formed at the eyepiece [5]. Some of the advantages are that they are compact and durable, closed tube design reduces image degrading air currents, they are excellent for lunar, planetary and binary star observing plus terrestrial viewing and photography, First-rate for deep sky observing or astrophotography with fast films or CCD's [5]. Some of the disadvantages are that they are more expensive than reflectors of equal aperture, Slight light loss due to secondary mirror obstruction compared to refractors.

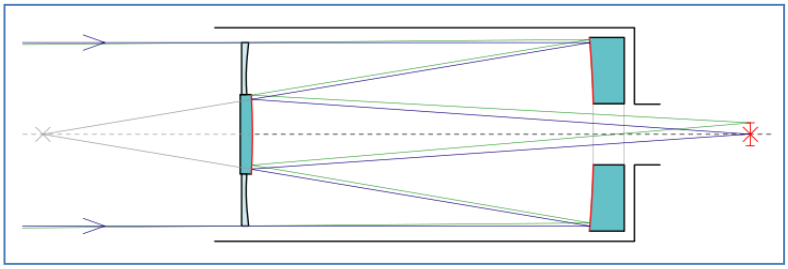


Figure 4: Light path is a Schmidt-Cassegrain [5]

2.2 Telescope Mounts (TM)

The primary purpose of any telescope mount is to accurately point the optics and imager at a desired target with repeatable accuracy. In the case of extended imaging (photographic or CCD), it must also track this target for the exposure duration,

countering the effects of the Earth's rotation and introducing as few inaccuracies as possible, thereby maximizing the quality of the data acquired [6].

The mount's adjustments should be smooth, yet precise, as you'll be using them to track the slow and steady apparent movement of celestial objects. Smooth and precise movements - and a motor drive - are an absolute requirement for astrophotography [7].

A telescope mount has two main functions:

1- Provide a system for smooth controlled movement to point and guide the instrument.

2- Support the telescope firmly so that you can view and photograph objects without having the image disturbed by tracking.

2.2.1 Current Telescope Mounts Types:

There are two major types of mounts for astronomical telescopes, namely: Altazimuth and Equatorial. They are explained in the following.

2.2.1.1 Altazimuth:

A mounting of this type is called an *alt-az* or *alt-azimuth* mount, as its two axes are aligned with respect to the altitude and azimuth axes of the horizontal coordinate system, as shown in Fig.5a. The tines of the fork are mounted on a rotating pad which acts as the azimuth axis. The altitude axis is formed by mounting both sides of the telescope near the top of the tines. One example of alt-azimuth mount is shown in Fig.5b which is the prototype designed in this thesis.

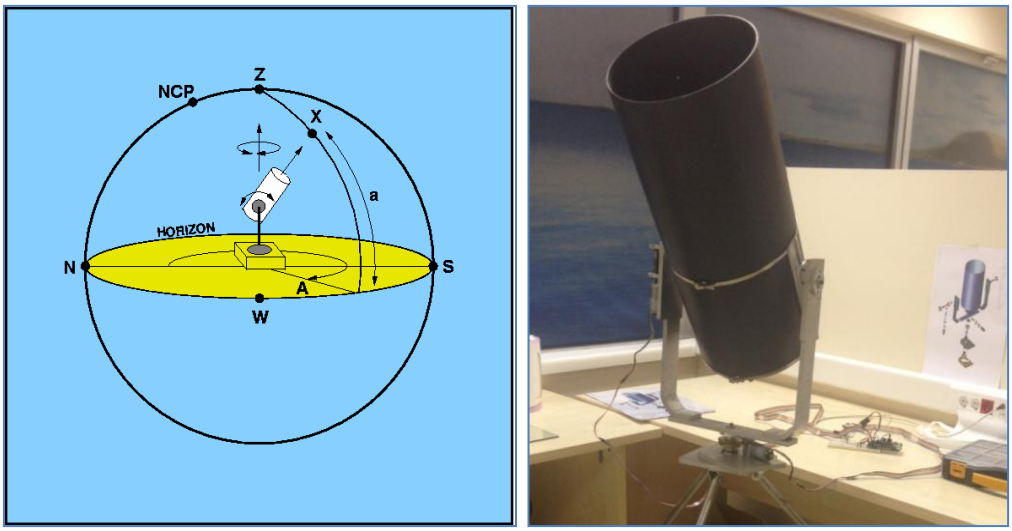


Figure 5a: Horizontal Coordinate System [8], b: Alt-Azimuth mount KAF-TM11

The primary disadvantage of the alt-azimuth design is that it is necessary to move both axes simultaneously in order to track the motion of an object as it moves across the sky due to the rotation of the Earth. Not only has this, but the speed that the axes have to be moved varied depending on where the object is in the sky. The calculations required to move the two axes are quite complex. There is a need for a coordinate conversion between the equatorial and horizontal coordinate systems and it needs to be executed many times in a second. This requires the use of a computer, but computers only became readily available in the 1970's. It is for this reason that every major research telescope made since the 1980's uses an alt-azimuth mount, but every telescope made before this time was mounted equatorially.

2.2.1.2 Equatorial:

This system is illustrated in Fig.6a. One axis is parallel to the Earth's rotation axis and is known as the *polar axis*. The second axis is perpendicular to this and is known as the *declination* (Dec) *axis*. The beauty of this design is that there is a direct relation between the equatorial coordinates of an object and the axes of the telescope.

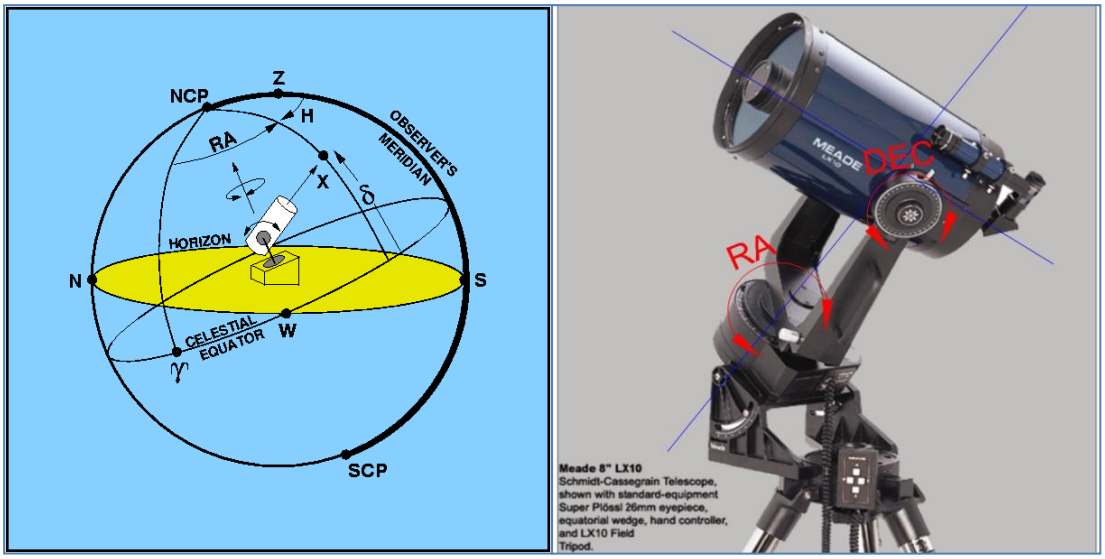


Figure 6a: Equatorial Coordinate System [8] and b: Equatorial mount sample

As well as making it easy to point to an object, this also makes it simple to track the motion of an object due to the Earth's rotation by driving just the *Right Ascension* (RA) *axis* at an equal angular velocity, but in the opposite direction, to the angular velocity of the Earth on its rotation axis [8]. An example of an equatorial mounting is shown in Fig.6b.

CHAPTER 3

CONCEPTUAL DESIGN:

In this chapter, we explain the concept of design of the telescope mount developed in this thesis. First of all the design criteria and requirements are listed.

3.1 Design Criteria and Requirements:

Design criteria and the requirements to be met by the telescope system are listed in the following.

- Size: OTA should be small class. Small class OTA aperture size; have the range from 4” to 16” (10 cm to 40 cm). OTA length should be maximum 100 cm. For this limitation preferred OTA should have possible Maksutov or Cassegrain types.

- Size: For Cassegrain type optics, focal plane instruments stay on back to the telescope optical tube. For this reason we have to consider 30 cm more space to back side of optical tube. Focal plane instruments can be CCD, filter wheel, focuser, eyepiece or all together.

- Loading capacity: Weight changes from 5 kg to 20 kg for the optical telescopes. We have to consider an extra weight of 15 kg takes into account the focal plane instruments weight. The weight limitations of the final design of the telescope mount altitude axis will be in between 20 kg to 35 kg. Telescope mount azimuth axis has to carry all optics in addition to the altitude mechanism. Telescope azimuth axis weight limitations are between 30 kg to 45 kg.

- Operating voltage: energy consumption should eliminate indoor or outdoor operation. Therefore 12 VDC voltages can be better choice.

- Tripod capacity: tripod top surface should stay parallel to ground. Tripod should have capacity to carry telescope.

- Both axes moving speed at the GOTO mode should be between 10deg/s to 3deg/s.

- Altitude range should be from 0° to 90°. Azimuth range should be from 0°- 360°.

- In tracking mode both axes tracking resolution should be better than 10 arc seconds.

- Existing telescopes were made iron or aluminum alloy. It has to be considered during material choosing stage.

3.2 Equipments:

The telescope system has two main parts, namely OTA and TM. The equipments of these parts are explained in the following sub-sections.

3.2.1 Optical Tube Assembly (OTA):

The subject of this thesis is to design a telescope mount. Therefore we planned to use the OTA developed by Aselsan Company (Fig.7). The specifications of the OTA are as follows:

- Optical Design: Cassegrain
- Aperture Size: 295cm (11")
- Focal Length: 2000 mm (78.7")
- Focal Ratio: f / 7.6
- Back Focal Length: 275 mm (10.8")
- Optical Tube: Delrin
- Optical Tube Length: 600 mm (23.6")
- Optical Tube weight: 6.80 kg (all in)

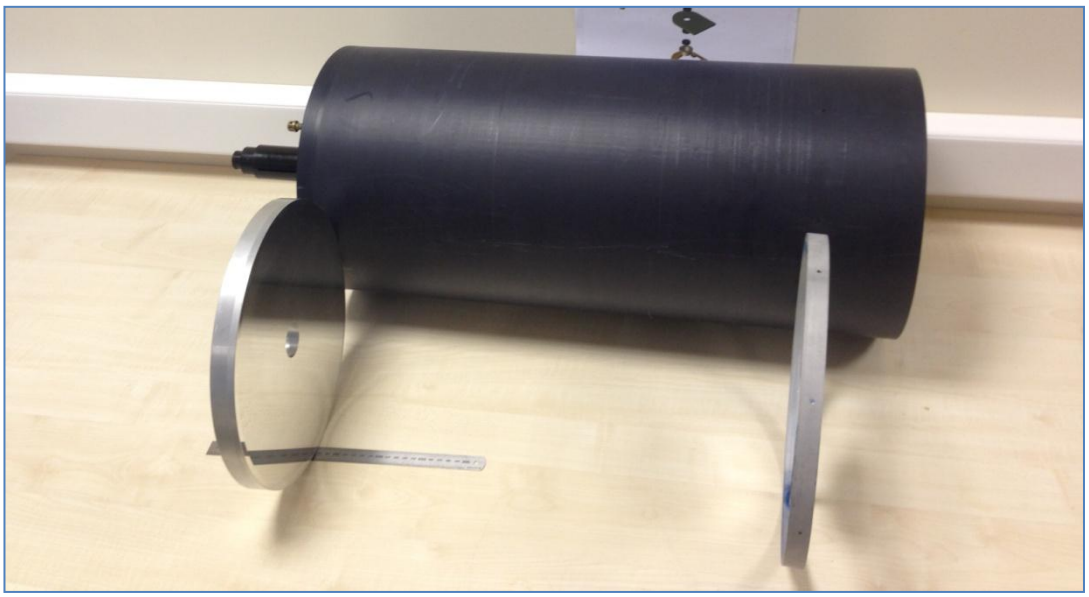


Figure 7: Optical Tube Assembly

The OTA is composed of a body and two mirrors as shown in fig.7. The body is a single piece delrin tube. There are two mirrors. The primary mirror has 295 cm diameter and the secondary mirror has 115 cm diameter.

3.2.2 Telescope Mount (TM)

TM consists of 2-degrees of freedom fork, driver electrical motors and control systems. In this thesis, all these subsystems are designed and a prototype system is manufactured. The design procedure is explained in the following sub-section.

3.3 Electro-Mechanical Model Design for Prototype:

The telescope system has both electrical and mechanical parts. The design procedure covers their selections based on the specifications, drawings for mechanical sub-systems and implementing control algorithms.

3.3.1 Mechanical Drawings:

Mechanical design has been done by using the well-known AutoCAD 14 drawing software. During design phase firstly customized equipments have been drawn. These equipments are DC motors, motor brackets, and ball bearings, worm and worm wheels.

3.3.2 DC motor selection:

DC motors are cheap and easily reachable choise. They have encoders and gear boxes. They can be controlled in a good and easy way. There may be some maintenance problems, but for the advantages listed above, we consider that they are applicable on this prototype. In our case, the speed required by the telescope is too low as compared to the nominal speed of the motor. Therefore we used worm gear to reduce the angular velocity and increase the torque required for the telescope system. This worm and worm wheel gear ratio is 1/34.

Desired target tracking angular velocity is equal to earth rotation angular velocity which is: ω_{tt}

$$\omega_{tt} = \frac{1^{\circ}}{240s} \text{ or } \omega_{tt} = \frac{\pi rad}{180 * 240s}$$

Desired target finding angular velocity should be between 10°/s and 3°/s which is: ω_{tf}
We intend to buy 12V geared DC motor that no-load velocity of $\omega_{no\text{-}load}$=80 rev/min. We did unit conversion for easy comparison as follows.

$$\omega_{no-load} = \frac{80rev}{min} * \frac{min}{60s} * \frac{360^{\circ}}{rev} * \frac{1}{34} = 14.11 \frac{^{\circ}}{s}$$

Or

$$\omega_{no-load} = \frac{80rev}{min} * \frac{min}{60s} * \frac{2\pi rad}{rev} * \frac{1}{34} = 0{,}24\frac{rad}{s}$$

$\omega_{no\text{-}load}$ is high from the desired value. But this problem can be handled with motor voltage adjustment. The specifications of the selected DC motor are given in the following sub-sections.

3.3.2.1 DC Motors:

The motors for the telescope system are 12 V brushed DC motors. They have dimensions of 69 x 37 x 37 mm in size. Gears of the motors have a reduction ratio of **131.25:1** with a metal gearbox. They have an integrated quadrature encoder that provides a resolution of 64 counts per revolutions of the motor shaft, which corresponds to **8384 counts per revolution** of the gearbox's output shaft. These units have D-shaped output shaft with a length of 15.5 mm, and a diameter of 6 mm. These motors were manufactured by Pololu Robotics & Electronics. Table 2 shows this motor and its important specifications. Altitude and azimuth axis movements are provided with these motors. Altitude and Azimuth axis DC motors are identical.

Table 2: Motor dimensions and general Specifications

Dimensions

Size:	37D x 69L mm
Weight:	8.1 oz = 229.6 g
Shaft diameter:	6 mm

General specifications

Gear ratio:	131.25:1
Free-run speed @ 12V:	80 rpm
Free-run current @ 12V:	300 mA
Stall current @ 12V:	5000 mA
Stall torque @ 12V:	250 oz·in = 9.0 kg.cm
Lead length:	27.94 cm

3.3.2.2 Motor Brackets:

This machined aluminum gear motor bracket lets us securely mount Pololu's 37D mm metal gearmotors to our project. The six holes on the bracket face line up with the six mounting holes on the 37D gearbox, and the bracket features

three tapped M3 holes on the bottom that can be used to mount it to a surface. The bracket includes nine M3 screws: three for mounting the bracket to a surface and six for mounting it to the gear motor [9].

3.3.3 Worm and Worm Wheels:

Worms and worm gears are gear sets that offer high gear reduction and torque multiplication. A worm drive is a cylindrical gear with a shallow spiral thread that engages the worm gear in a non-intersecting, perpendicular axes configuration.

Efficiency of a worm drive is determined by the lead angle and the number of threads in contact with the worm gear. A high lead angle on the drive reduces frictional losses and heat. A low lead angle reduces gear speed while proportionally increasing torque [24].

Friction is an issue with all worm sets; the worm gear cannot transfer motion back to the worm drive in most instances [24]. Bronze toothed wheel and a ground stainless steel screw worm material used for the both axis motions. The worm and worm wheel gear ratio is 1/34 and these sets were shown in Fig.8.

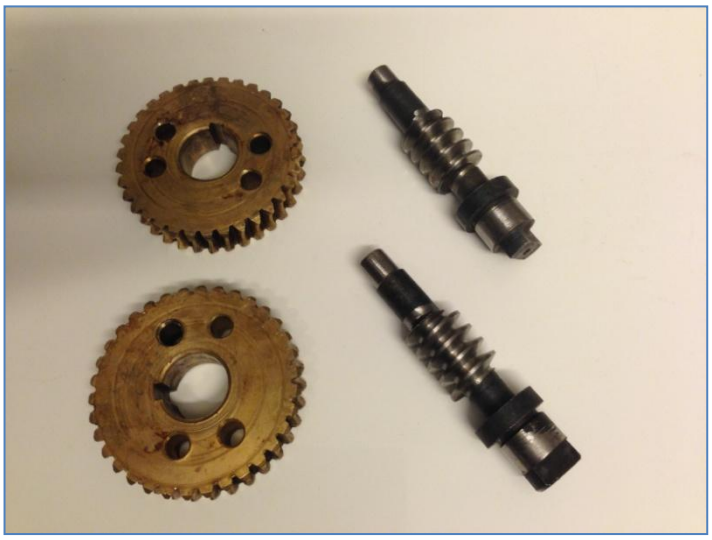


Figure 8: Worm and Worm Wheel

Backlash, a clearance between mating gear teeth, is built into speed reducers to let the gears mesh without binding and to provide space for a film of lubricating oil between the teeth. This prevents overheating and tooth damage [10].

On the other hand, the same clearance causes lost of motion between reducer input and output shafts, making it difficult to achieve accurate positioning in equipment such as instruments, machine tools, and robots [10]. Backlash comes from our worm and worm wheel will determine and find total effects on the position accuracy.

3.3.4 Ball Bearings:

Ball bearings are used on both axes of the prototype for reducing the frictions and increasing the precision. Two single row, stainless steel, and deep grove ball bearings are using on both fork arms for altitude movements. Two single row, stainless steel, and deep grove ball bearings and one stainless steel, and deep grove tapered roller bearing are used on azimuth shaft for azimuth movements. Mounting bearings to support a shaft are that the bearings should be spaced 3-5 shaft diameters apart if the bearings are to effectively resist moments applied to the shaft. Ball bearing's pictures and sections are shown on.Fig.9 [11].

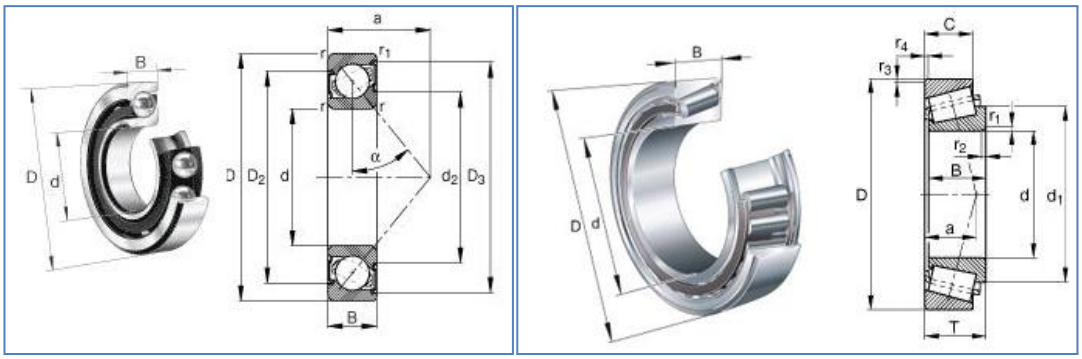


Figure 9: Ball bearing's pictures and sections [11]

3.3.5 Main Body:

In the process of mechanical identification of the system the telescope azimuth and elevation models are determined separately. The models are independent of each other because there is a very weak coupling between the axes. However, the azimuth model depends on the elevation position of the telescope. The telescope structural properties, as measured at the elevation axis, do not depend on the elevation position. Thus, the elevation structural modes and frequencies are also independent on the telescope elevation position [12].

A similar situation is observed for azimuth models. Properties of the azimuth model do not depend on the telescope elevation position because the telescope altitude accepted on axis and weight is balanced.

Main body design was started after above equipments 3D models were ready. Body structure has designed three main parts.

1. Altitude axis structure
2. Azimuth axis structure
3. Tripod

3.3.5.1 Altitude axis structure

The mass of movement on altitude axis was identified as M1. M1 comes up with ball bearings, shafts, worm wheel, OTA connection skidders, OTA, and focal plane instruments. It is necessary to balance all equipments on altitude axis to reduce motor loads. Ball bearings, altitude shafts, and worm wheel are already on altitude axis which we identified X axis. But the center of mass of the OTA together with the focal plane instruments is calculated here. The center of mass of these parts is brought to X axis by the help of skidders.

Under favour of fork arm mount M1 weight is divided by two and M1 balanced on the Y axis also. Therefore it was used motor with a smaller torque for altitude axis. M1 altitude axis structure was shown on Fig.10.

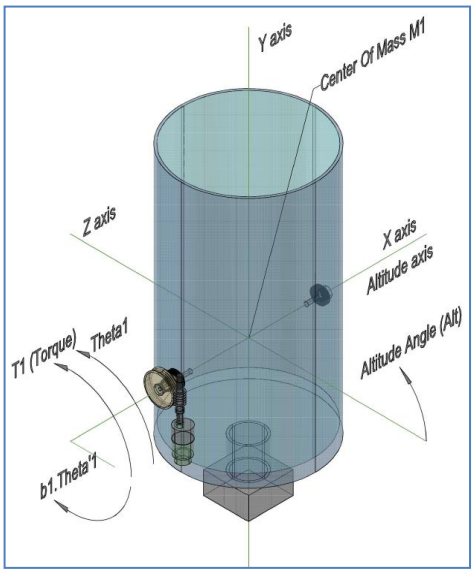


Figure 10: M1 Altitude Axis Structure

3.3.5.2 Azimuth axis structure

The mass of movement on azimuth axis was identified as M2. M2 comes up with total mass of M1, ball bearings, azimuth shaft, fork arms, worm wheel, and altitude motors. It is necessary to balance all equipments on azimuth axis for reducing the motor loads. Ball bearings, azimuth shaft, worm wheel, and fork arms are already on azimuth axis which we identified as Y axis.

Alt-Az mount has advantages which mass of M1 carries by tapered roller bearing and ball bearings on Y axis. Therefore it was used motor with a smaller torque for azimuth axis movements. M2 azimuth axis structure was shown on Fig.11.

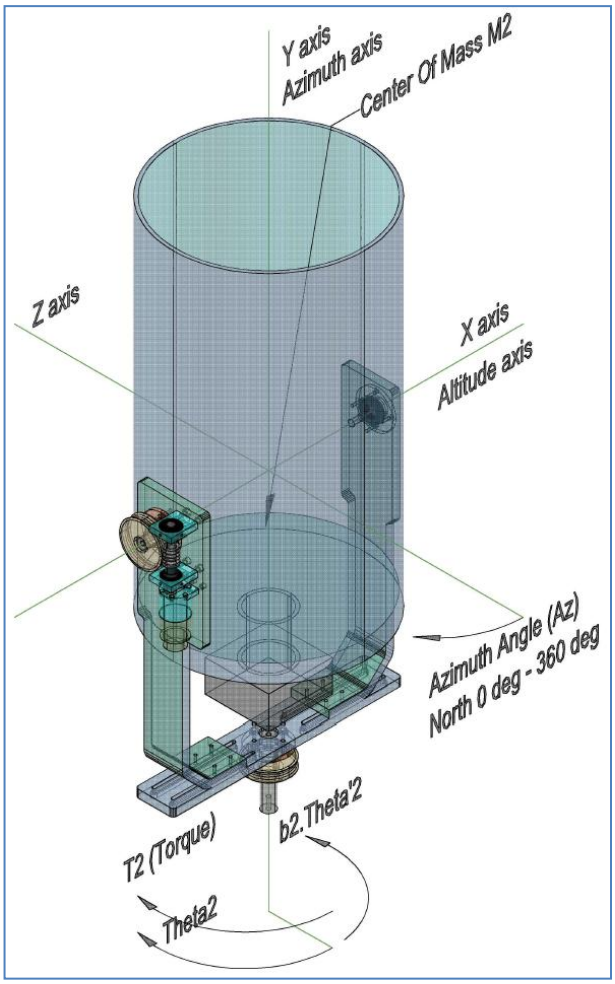


Figure 11: M2 Azimuth Axis Structure

The Fork set up is made of two 12 mm thick arms with a U shaped section. One arm is provided with a screw worm reduction system used to drive the telescope, the other one is provided with a ball bearing having a slight axial play to compensate possible expansion. We didn't have an OTA when we design stage. Therefore we designed fork arms distance variable.

The distance between the altitude axis and the fork arm base is 400 mm which gives the opportunity of mounting a wide range of telescopes with their photographic instruments.

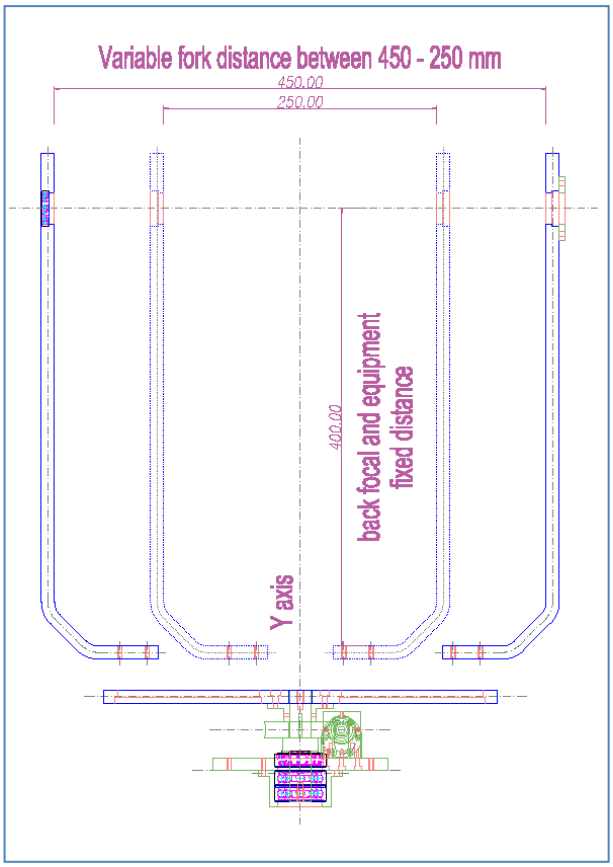


Figure 12: Variable Fork Arms

3D designs are giving mechanical properties for M1 and M2 masses. Table 3 has shown these properties.

Table 3: Mechanical properties for M1 and M2 Masses

Mass:	**M1: 4961 g**	**M2: 6109 g**
Bounding box:	X: -176.0000 -- 202.8000	X: -208.4000 -- 180.0000
	Y: -305.0000 -- 375.0000	Y: -60.0000 -- 775.0000
	Z: -150.0000 -- 150.0000	Z: -150.0000 -- 150.0000
Centroid:	X: 4.8391	X: -5.7192
	Y: -40.2619	Y: 327.0971
	Z: -0.5817	Z: 0.9794
Moments of inertia:	**X: 2.2129E+11 g.mm^2**	X: 9.2073E+11
	Y: 79849359623.5984	**Y: 1.0257E+11 g.mm^2**
	Z: 2.2614E+11	Z: 9.4720E+11
Products of inertia:	XY: -1054767374.1372	XY: -1.2561E+10
	YZ: 216072836.7229	YZ: 2068087227.0840
	ZX: -535516932.8642	ZX: -1049641330.1764
Radii of gyration:	X: 211.1853	X: 388.2012
	Y: 126.8583	Y: 129.5687
	Z: 213.4888	Z: 393.7408

Tolerances of parts were given properly, when parts were designed. Every parts shop drawing were done separately. All shop drawings are finding at Appendix B. Exploded and assembled views of the telescope mount are shown Fig.13.a and Fig.13.b respectively.

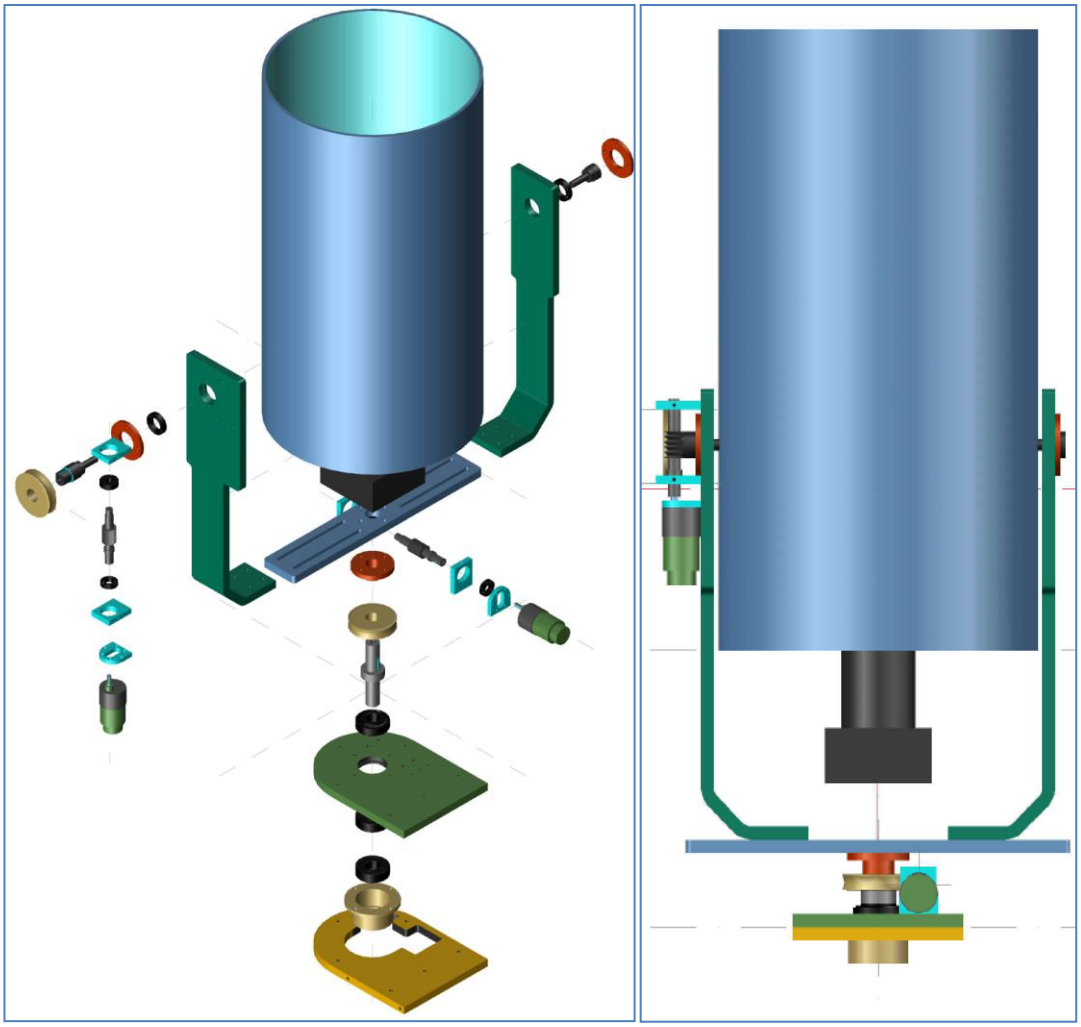


Figure 13.a: Telescope mount design's exploded view, b: Telescope mount design's assembled view.

CHAPTER 4

CONTROL SYSTEM

4.1 Control system design

The system has 2-degrees of freedom. One of them is through the azimuth-axis and the other is through the altitude-axis. We have to control both of the axes with zero steady-state error to desired step inputs. The controllers to be used in both azimuth-axis and altitude axis are selected as the well-known Proportional-Integral-Derivative (PID) controller [13] [14] [15].

As we know that the P-control speeds-up the response. Since there may be a steady-state error while using only P-Control, we added an Integral-Control to the system based on our requirements for steady-state error. The D-Control is not essential for our system since the response is relatively slow when we move the telescope to a desired target star or planet. Therefore, during simulations and actual operations, we used PI-Control. The block diagram of controlling the telescope in one of the axis is given in Fig.14 in s-domain. It shows the analog PID control system; however, since we intended to use a digital computer to implement the controller, the control system is transformed into the discrete control system as shown in Fig.15 in z-domain. Normally, our control system is a sampled-data control system since the plant to be controlled is a DC motor which is naturally an analog system [16].

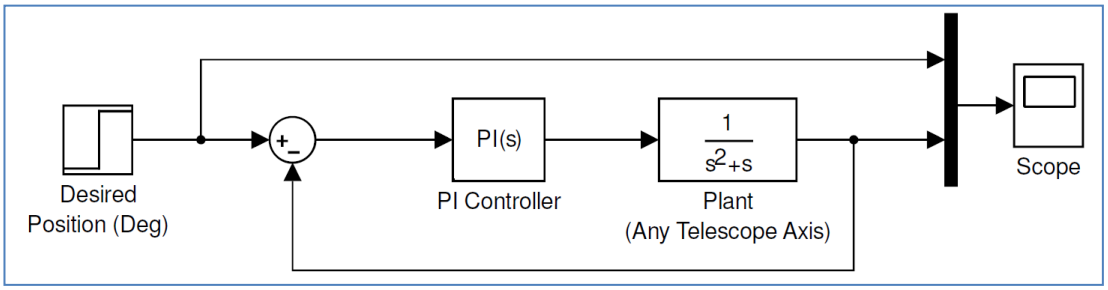


Figure 14 Analog Control System in S-Domain

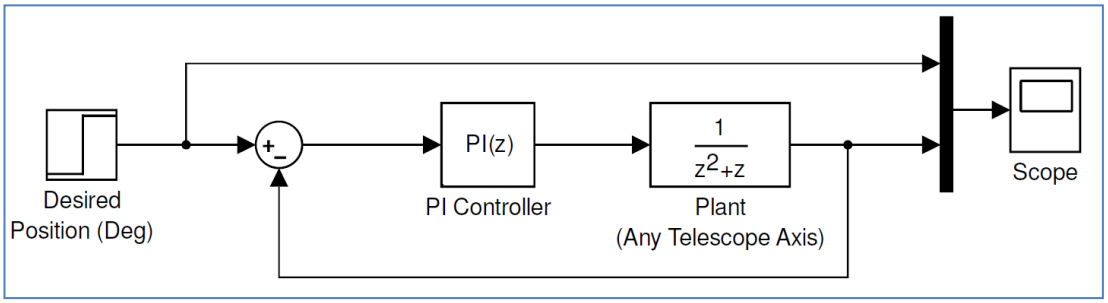


Figure 15 Sampled-Data Control System in Z-Domain

The parameters P, I and D can be determined based on the mathematical model of the overall system. They may be a need for fine-tuning based on some experiments. But, in order to obtain reasonable values for PID parameters, we also wanted to use a system identification process as explained later in the thesis. In normal operations, we take the gain parameter of D-controller as zero.

Fig.16 depicts the overall system block diagram. It gives the components of the telescope mount (TM). The physical components of the TM are explained in the following sub-sections.

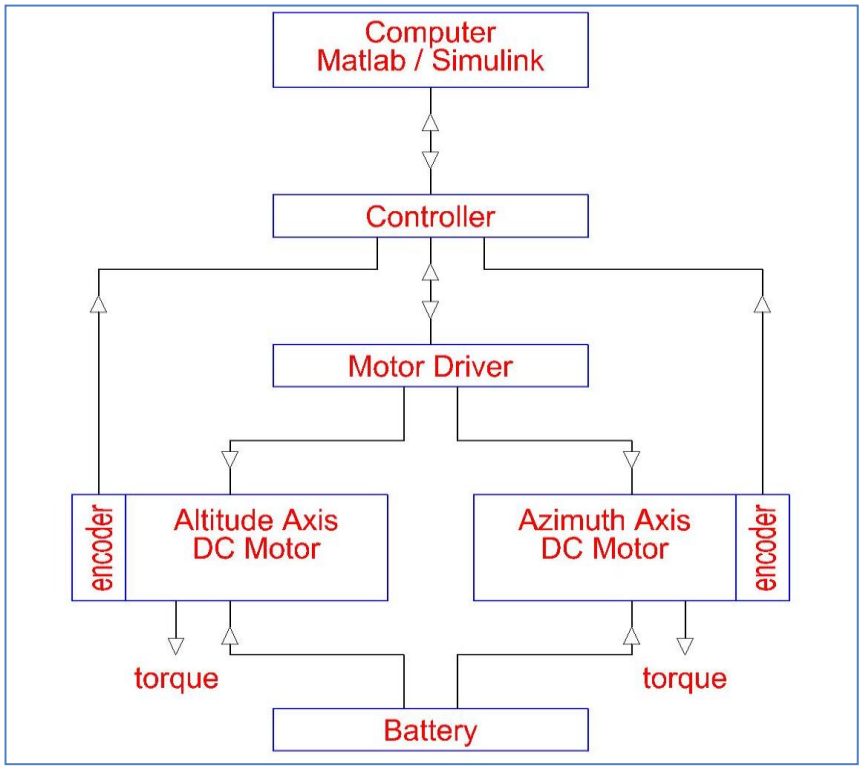


Figure 16: Control System

4.1.1 Main controller PC

The PID controllers are implemented in a personal computer (PC) which has Intel core i7-3740QM CPU, 16 GB ram, Quadro K3000M screen card, MS windows 7 64-bit operating system. PID controllers are implemented by Matlab/Simulink software for real time system control. Arduino mega 2560 control board and Pololu dual VNH5019 motor driver shield softwares are also available as the Simulink modules.

4.1.2 Arduino Mega 2560 Micro Controller Board

As it is explained in the website of the Arduino, the Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins

(of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started [17].

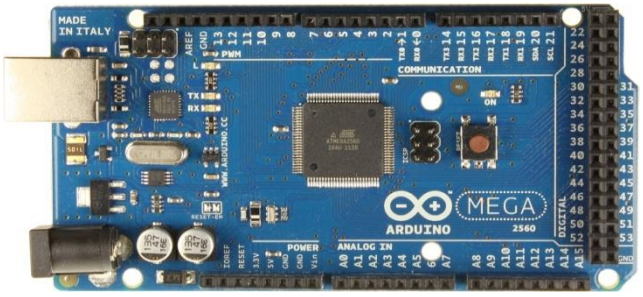


Figure 17: Arduino Mega 2560 Controller

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data.

- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details [17].

- PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analog Write () function [17].

4.1.3 Dual Motor Driver:

As it explained in its website, the Pololu dual VNH5019 motor driver shield for Arduino and its corresponding Arduino library make it easy to control two bidirectional, high-power DC motors with an Arduino or Arduino clone. The board features a pair of robust VNH5019 motor drivers from ST, which operate from 5.5 to 24 V and can deliver a continuous 12 A (30 A peak) per channel, and incorporates most of the components of the typical application diagram on page 14 of the VNH5019 datasheet which you can reach www.pololu.com [18] address, including pull-up and protection resistors and FETs for reverse battery protection [18].

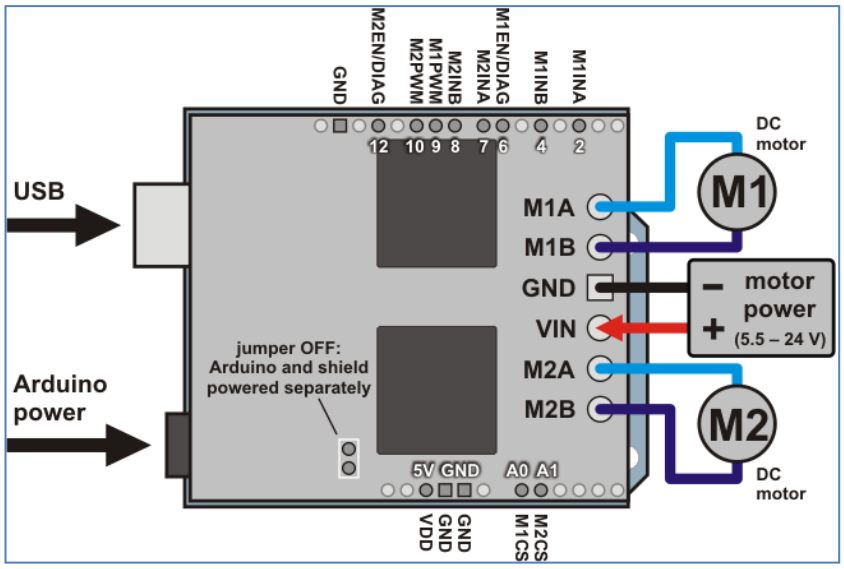


Figure 18: Dual VNH5019 Motor Driver Shield with an Arduino [18]

All of the necessary logic connections between the Arduino and the motor driver shield are made automatically when the shield is plugged into the Arduino. However, the shield's power connections must be made directly to the shield itself via its large VIN and GND pads. The Fig.18 shows the typical connections involved in using this board as an Arduino shield [18].

4.1.4 DC Motors:

The motors used to drive the TM in both azimuth-axis and altitude-axis are brushed DC motors. The operating voltage of the motors that we selected is 12V. Each motor has a gearbox with a reduction ratio of 131:1 (Fig.19). It has also a quadrature encoder for feedback application.

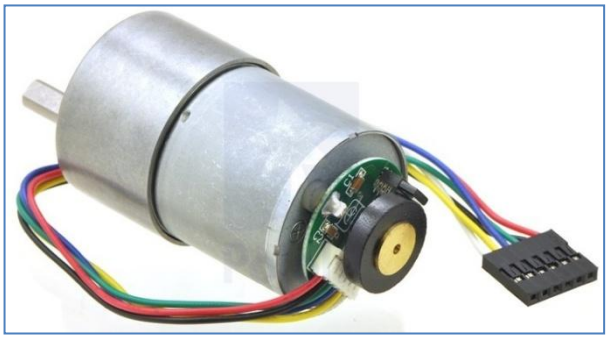


Figure 19: 12V Brushed DC Motor, a 131:1 Gearbox and a Quadrature Encoder [18]

4.1.5 Battery:

Based on the needed supplying voltages and loading current of the motors, IR sensors and electronic boards, it's decided to use a battery with the voltage, 11.1v and power, 5200 mAh. Fig.20 shows the used battery.

Figure 20: 5200 mAh Battery Pack

4.2 Electromechanical System Transfer Functions and Model Design:

The electrical model of the DC motor is shown in Fig.21.

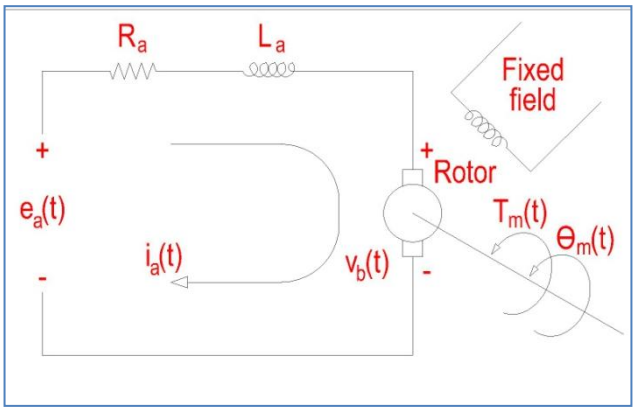


Figure 21 The Electrical model of the DC Motor [14]

The back emf, $v_b(t)$,is related to the angular displacement $\theta_m(t)$of the rotor as in Eq.1.

$$v_b(t) = K_b \frac{d\theta_m(t)}{dt} \qquad (1)$$

Where, K_b is the constant of back emf. Derivative of the angular displacement is the angular velocity $\omega_m(t)$ of the rotor as in Eq.2.

$$\frac{d\theta_m(t)}{dt} = \omega_m(t) \qquad (2)$$

Laplace transform of the Eq. 1 is;

$$V_b(s) = K_b s\theta_m(s) \qquad (3)$$

If we write a loop equation for the armature circuit in Fig.21, we find the relationship between armature current $i_a(t)$, applied armature voltage $e_a(t)$ and the back emf $v_b(t)$ in s-domain as in Eq.4.

$$R_a I_a(s) + L_a s\, I_a(s) + V_b(s) = E_a(s) \qquad (4)$$

The motor torque is proportional to the armature current,

$$T_m(s) = K_t I_a(s) \quad or \quad I_a(s) = \frac{1}{K_t} T_m(s) \qquad (5)$$

The transfer function we have to substitute eq. (3) and (5) into (4),

$$\frac{(R_a + L_a s) T_m(s)}{K_t} + K_b s \theta_m(s) = E_a(s) \qquad (6)$$

The relation between the torque $T_m(s)$ and the angular displacement $\Theta_m(s)$ is given in Eq.7. Equivalent typical mechanical loading on the motor can be seen in Fig.20.

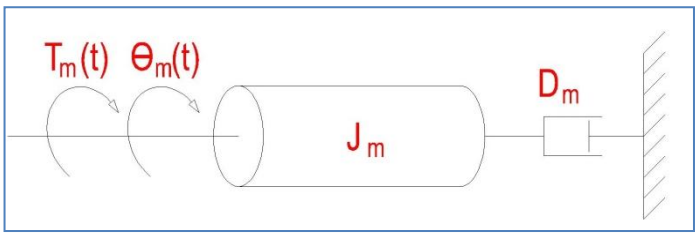


Figure 22: Equivelent typical mechanical loading on a motor [14]

$$T_m(s) = (J_m s^2 + D_m s)\theta_m s \qquad (7)$$

Where, J_m is the moment of inertia and D_m is the viscous damping coefficient. If we substitute Eq. (7) into Eq. (6):

$$\frac{(R_a + L_a s)(J_m s^2 + D_m s)\theta_m s}{K_t} + K_b s \theta_m(s) = E_a(s) \qquad (8)$$

We can assume that the armature inductance L_a is equal to zero, since it is too small in compare with the armature resistance R_a. Then Eq.8 can be simplified and written as in Eq. (9).

$$\left[\frac{R_a}{K_t}(J_m s + D_m) + K_b\right] s\theta_m(s) = E_a(s) \qquad (9)$$

The transfer function $\theta_m(s)/E_a(s)$ is obtained as in Eq. (10).

$$\frac{\theta_m(s)}{E_a(s)} = \frac{\frac{K_t}{R_a J_m}}{s\left[s + \frac{1}{J_m}(D_m + \frac{K_t K_b}{R_a})\right]} \qquad (10)$$

If we define the numerator coefficient as $K=K_t/(R_aJ_m)$ and denominator coefficient as $\alpha=(1/J_m).(D_m+K_tK_b/R_a)$, we can simplify the transfer function as in Eq. (11).

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K}{s[s+\alpha)]} \qquad (11)$$

The relationships among the mechanical constants and motor and gear ratios are given in Eq.12 (See also Fig.23).

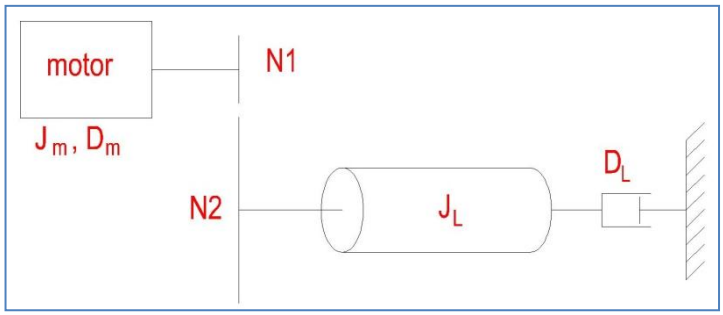


Figure 23: Relationships between the motor, gear and load [14]

$$J_m = J_a + J_L\left(\frac{N_1}{N_2}\right)^2 \quad ; \qquad D_m = D_a + D_L\left(\frac{N_1}{N_2}\right)^2 \qquad (12)$$

Where m, a and L stand for motor, armature and load respectively.

Taking the inductance $L_a=0$, eq. (6) becomes;

$$\frac{R_a}{K_t} T_m s + K_b s\theta_m(s) = E_a(s) \qquad (13)$$

And taking inverse Laplace transform of eq. (13);

$$\frac{R_a}{K_t} T_m(t) + K_b \omega_m(t) = e_a(t) \qquad (14)$$

Equation (14) says if constant voltage e_a is applied to the dc motor, it will turn in a constant angular velocity with constant torque T_m.

$$\frac{R_a}{K_t} T_m + K_b \omega_m = e_a \qquad (15)$$

Take ω_m to one side of eq. (15) and so;

$$T_m = -\frac{K_b K_t}{R_a}\omega_m + \frac{K_t}{R_a}e_a \qquad (16)$$

T_m and ω_m relations are shown on Fig.24 which is the *torque – speed curve* of the DC motor.

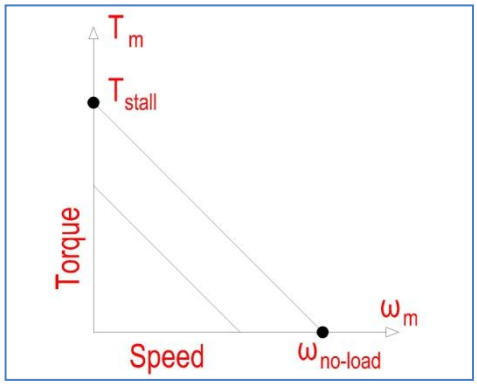


Figure 24: Torque – Speed curve [14]

Fig.24 plots show us when the angular velocity reaches zero, that value of torque is stall *torque* T_{stall}. And when the torque is zero, that value of angular velocty is *no-load speed*$\omega_{no-load}$. These are shown in eq. (17) and eq. (18) [14].

$$T_{stall} = \frac{K_t}{R_a}e_a \qquad or \qquad \frac{K_t}{R_a} = \frac{T_{stall}}{e_a} \qquad (17)$$

$$\omega_{no-load} = \frac{e_a}{K_b} \qquad or \qquad K_b = \frac{e_a}{\omega_{no-load}} \qquad (18)$$

Now we can apply that driven formulas to our representative altitude axis structure and azimuth axis structure one by one (Table 4).

Table 4: Altitude and azimuth axis structure

model design constants	**altitude structure**	**azimuth structure**	**values provided from**
J_a: motor moment of inertia (Kg.m^2)	0,034	0,034	motor specs
D_a: motor viscous damping coeff. (N.m.s/rad)	0,375	0,375	motor specs
J_L: Load moment of inertia (Kg.m^2)	221,29	102,57	3D model
D_L: Load viscous damping coeff. (N.m.s/rad)	400,00	600,00	similar app.
T_{stall}: Stall Torque (Kg.m)	0,09	0,09	motor specs
$W_{no-load}$: No Load Speed (rad/s)	0,24	0,24	motor specs
e_a: voltage (V)	12,0	12,0	motor specs
N_1: worm gear tooth	1	1	it self
N_2: worm wheel tooths	34	34	it self

Table 5 shows the constants for the model. For the Simulink model, we have to calculate J_m, D_m, K_t/R_a, and K_b values according to table 4. We use eq. (12), (17) and (18) for calculations.

Table 5: Constants for Models

Calculation model design constants	**altitude structure**	**azimuth structure**	**values provided from**
J_m: Total moment of inertia (Kg.m^2)	0,225	0,123	Eq.(12) & Table-4
D_m: Total viscous damping coeff. (N.m.s/rad)	0,721	0,894	Eq.(12) & Table-4
K_t/R_a: stall torque voltage relation coeff.	0,01	0,01	Eq.(17)
K_b: voltage no load speed relation coeff.	50,00	50,00	Eq.(18)

Putting these values into our telescope mount system models in Simulink, it gives the following transfer functions, $G_1(s)$ and $G_2(s)$ representing to altitude axis structure and azimuth axis structure respectively. "1" subscript represents altitude values and "2" subscript represents azimuth values.

$$G_1(s) = \frac{\theta_{m1}(s)}{E_{a1}(s)} = \frac{\frac{K_{t1}}{R_{a1}J_{m1}}}{s\left[s + \frac{1}{J_{m1}}(D_{m1} + \frac{K_{t1}K_{b1}}{R_{a1}})\right]} \tag{19}$$

$$G_2(s) = \frac{\theta_{m2}(s)}{E_{a2}(s)} = \frac{\frac{K_{t2}}{R_{a2}J_{m2}}}{s\left[s + \frac{1}{J_{m2}}(D_{m2} + \frac{K_{t2}K_{b2}}{R_{a2}})\right]} \tag{20}$$

4.3 System model representation with Simulink:

The overall telescope mount system is implemented in Simulink as shown in Fig.25. As seen, the upper part of the figure is the altitude control system and the lower part is the azimuth control system. The controllers for both systems are PID controllers.

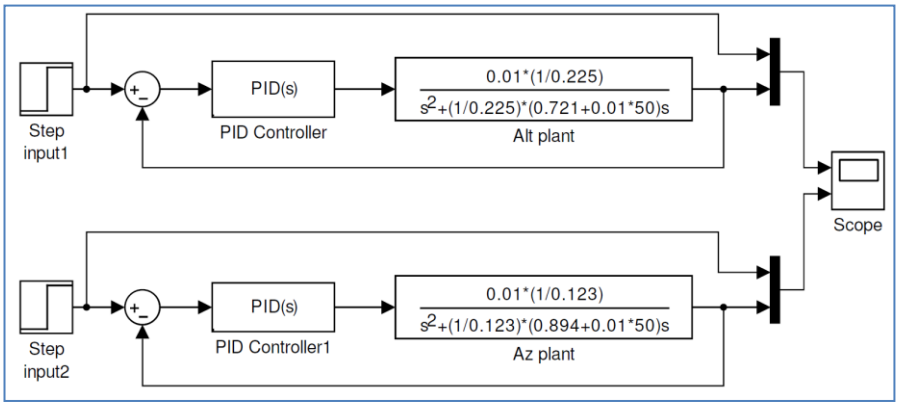


Figure 25: System mathematical model representation

We apply to step input for both axis. The coefficients in the altitude and azimuth transfer functions were taken from Table 5.

First of all, we applied P controller to our system. We used Matlab PID tuners. Matlab PID tuner gives us a setting space for P values. We can adjust system respons time or transient behavior by changing the PID gain parameters. Fig. 26.a.b.c shows the responses for system where only P controller was applied.

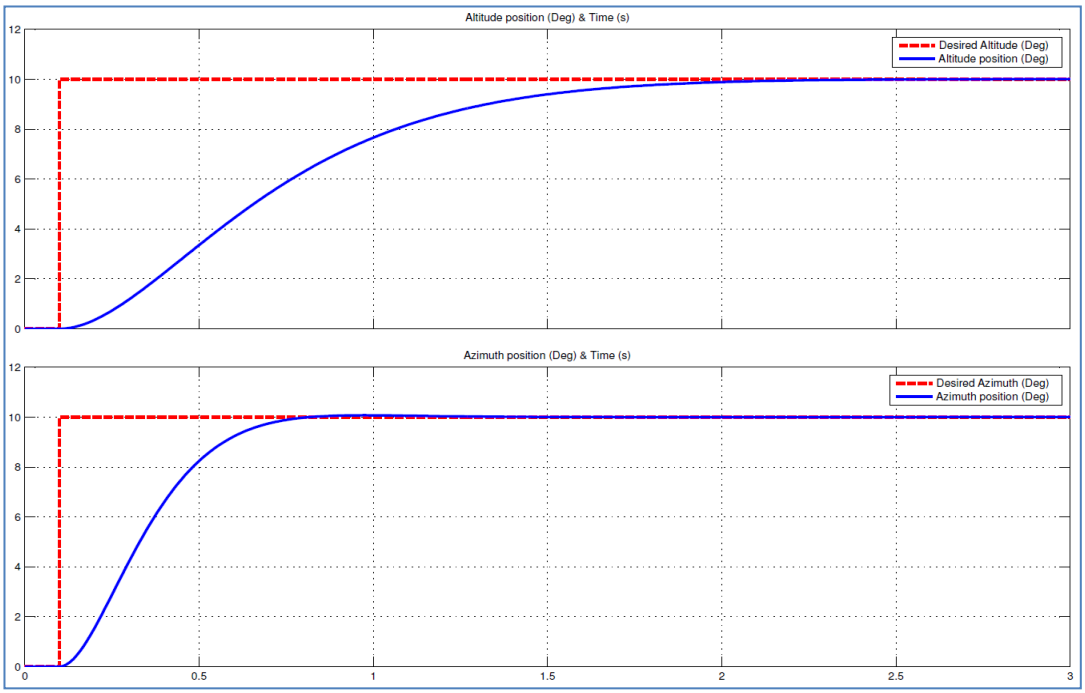


Figure 26.a: P Controller response for K_P=190 and K_P=530

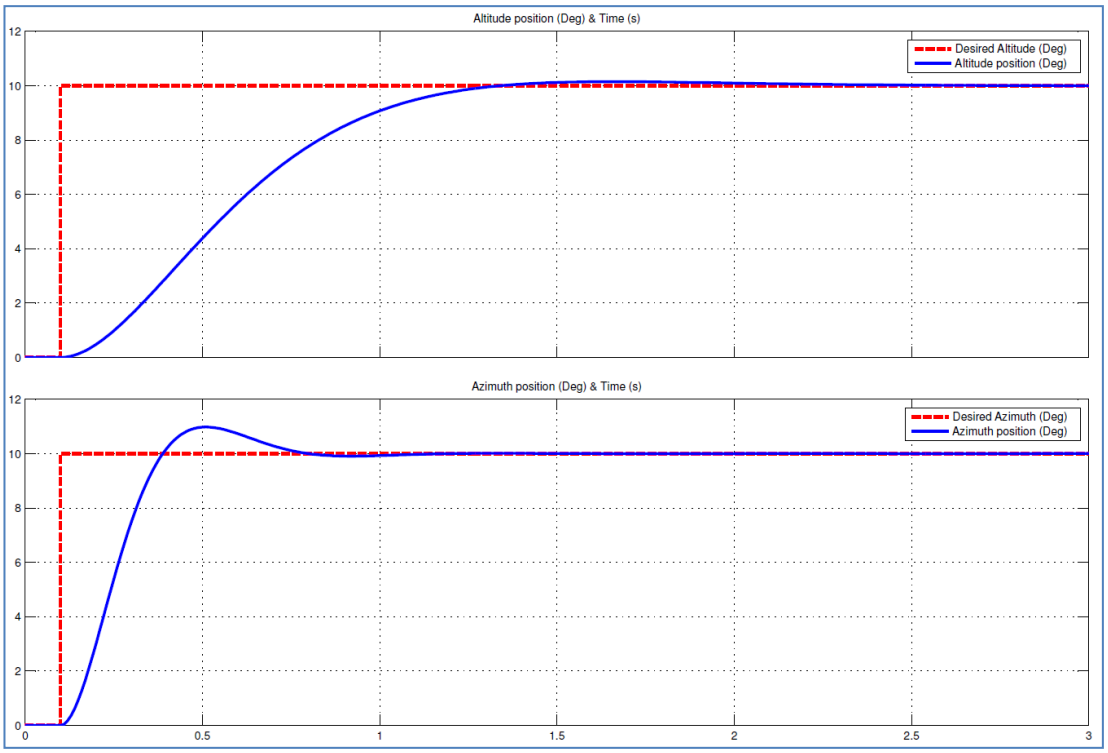


Figure 26.b: P Controller response for K_P=256 and K_P=1116

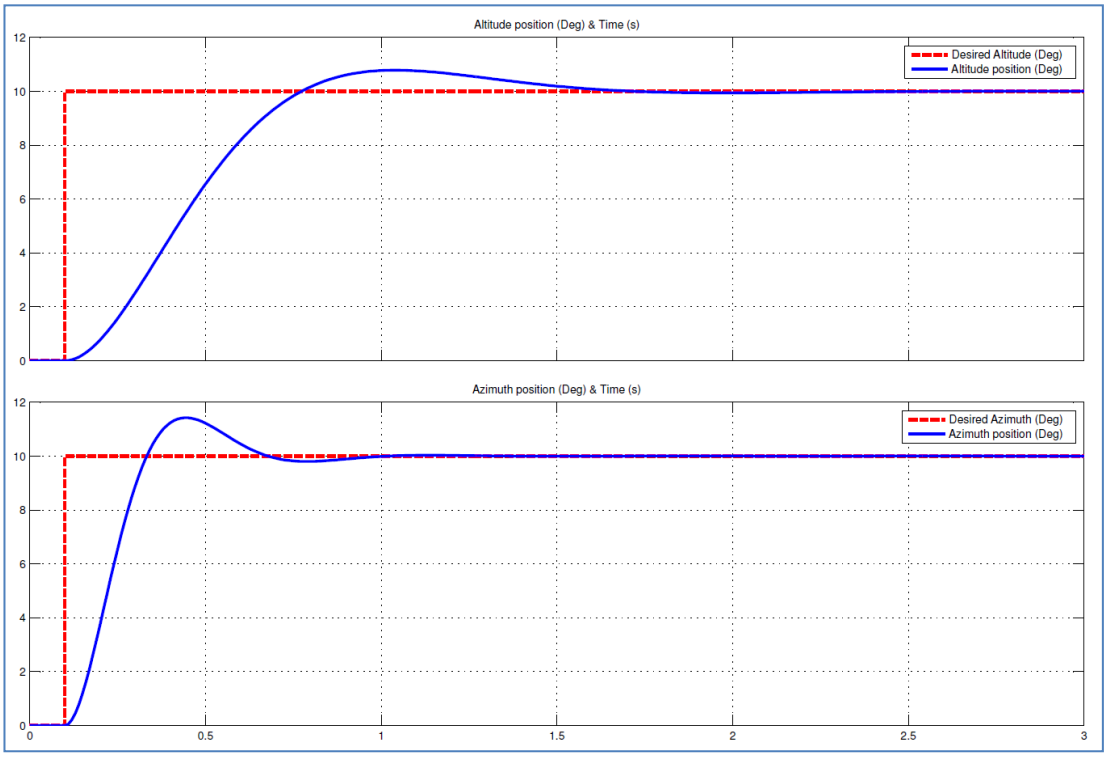


Figure 26.c: P Controller response for K_P=417 and K_P=1419

Also we applied PID controller to both system with the same way. Systems responses with PID controller are shown Fig.27.a.b.c. It shows the responses for system where only PID controller was applied. In these figures shows different K_P,

K_I, K_D values. During PID tunes; K_P, K_I, K_D values were manipulated to aim of the system more robast and faster.

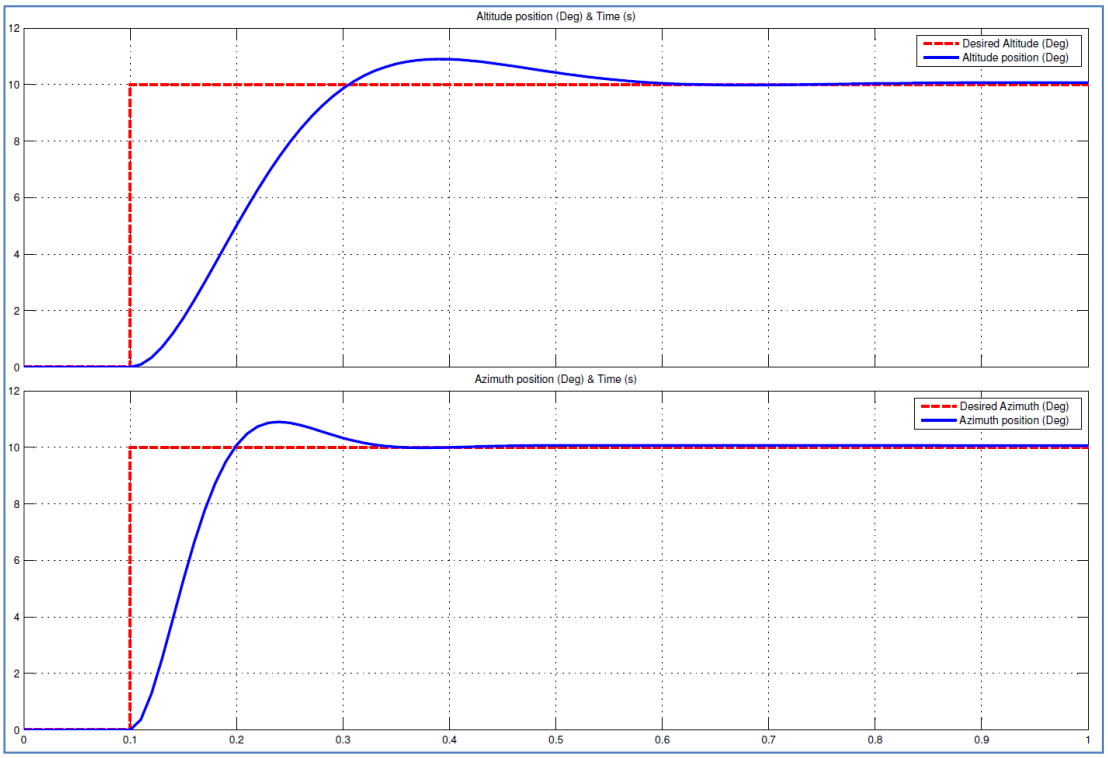


Figure 27.a: PID Controller response for K_P=1351, K_I=113, K_D=171 and K_P=3221, K_I=562, K_D=196

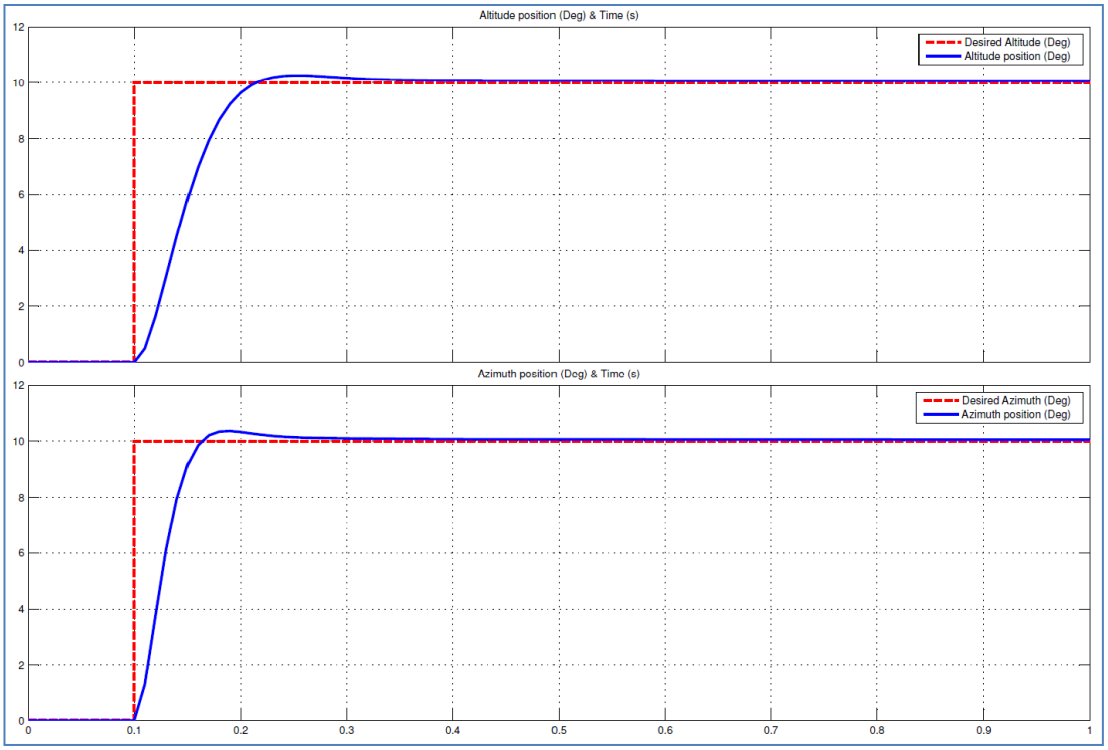


Figure 27.b: PID Controller response for K_P=2742, K_I=471, K_D=433 and K_P=5677, K_I=1705, K_D=396

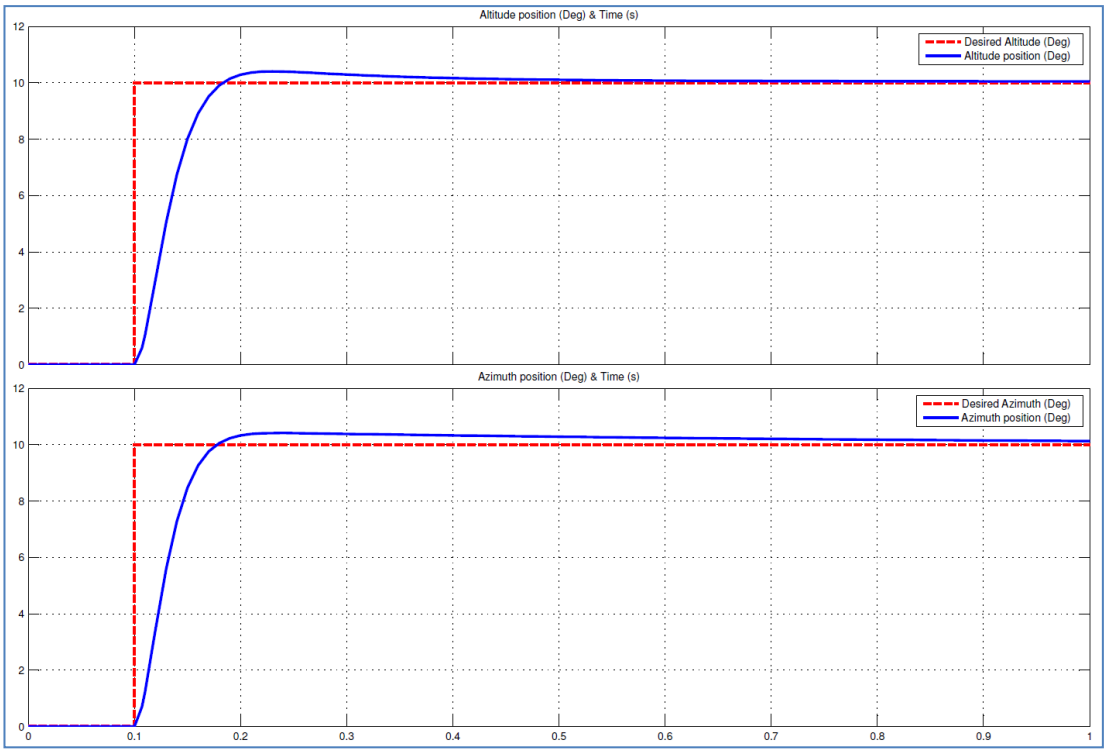


Figure 27.c: PID Controller response for K_P=4669, K_I=1103, K_D=596 and K_P=4907, K_I=6427, K_D=345

Of course we did some assumptions and ignorance's when we were creating to model. If the effects that these ignored properties on the response are small, good agreement will be obtained between the results of the analysis of a mathematical model and the results of the experimental study of the physical system [15].

Now its time to start prototyping our telescope mount with respect to mechanical design and mathematical models.

CHAPTER 5

PROTOTYPE DEVELOPMENT AND CONSTRUCTION

We explained our design phase and design jobs in previous sections. In this chapter, we explain prototyping aspects of mechanic structure, electronics and software of the telescope.

5.1 Production and assembling of mechanic components

Manufacturing and purchasing stages have started after the design phase have completed. Motors, motor brackets, ball bearings were purchased. Because of its light and easy handling properties, we decided to use aluminum alloy material to manufacture mechanic structure. Aluminum alloy type and production code is 5083.

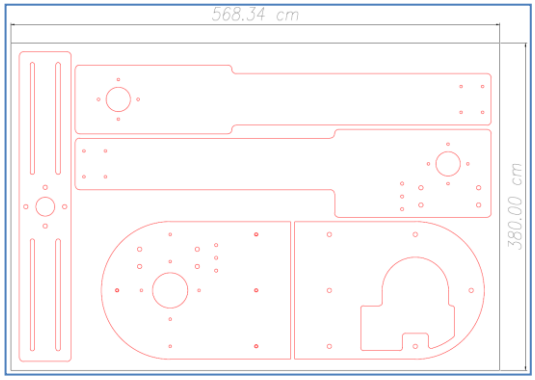


Figure 28: 12 mm Thickness plate size and manufacturing model

Except cylindrical parts, we bought two aluminum alloy plates. The thickness of the first one is 8 mm and the other is 12 mm. 12 mm thickness plate size and manufacturing model are shown Fig.28. Water jet cutter has been used for cutting these metals. After these cutting operations results are shown at Fig.29.

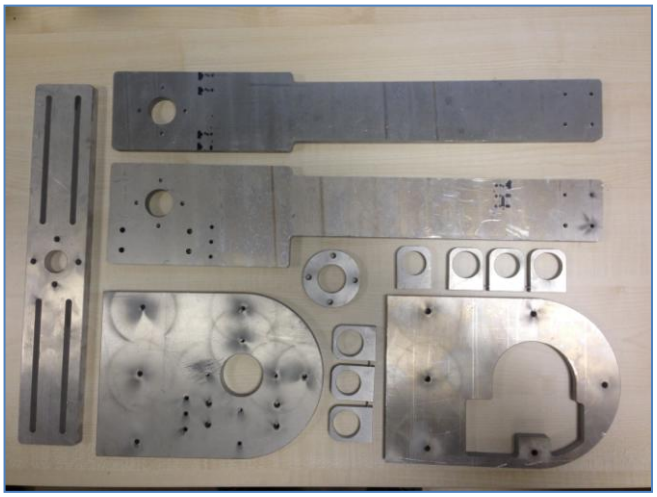


Figure 29: The parts after cutting operations

Manufacturing, mount and assembling operations are all done at Atılım University mechanic workplace. Fig.30, 31 and 32 show the assembling process.

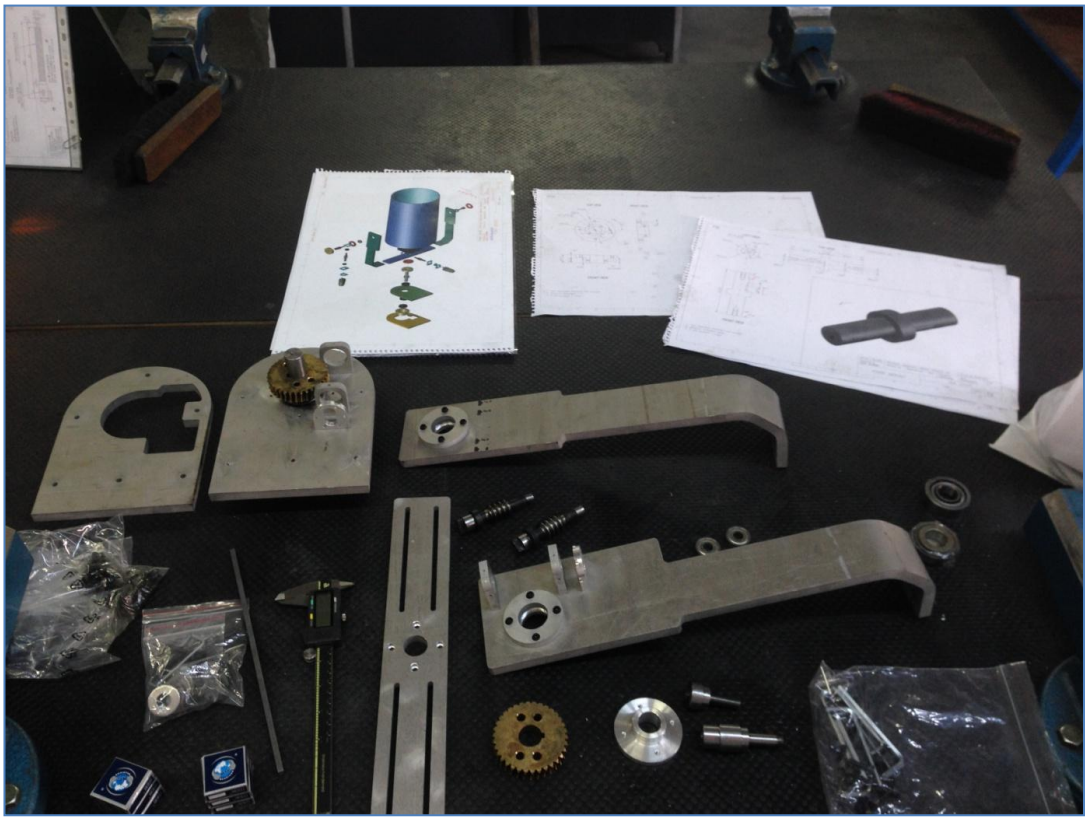


Figure 30: At the beginning of the assembling phase

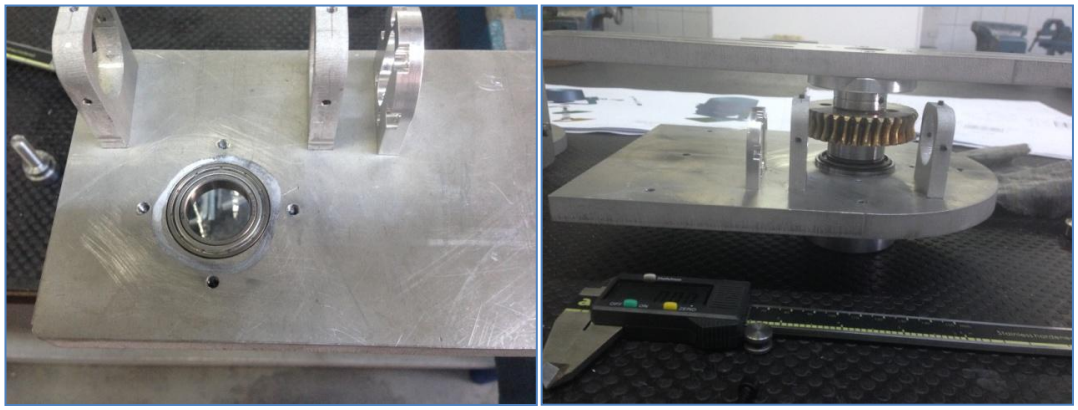


Figure 31a and b: Some of details of mount phase

Figure 32: Completed Telescope Mount and Telescope Optics

5.2 Electronics and software implementation:

Telescope control system structure is shown in Fig.33.

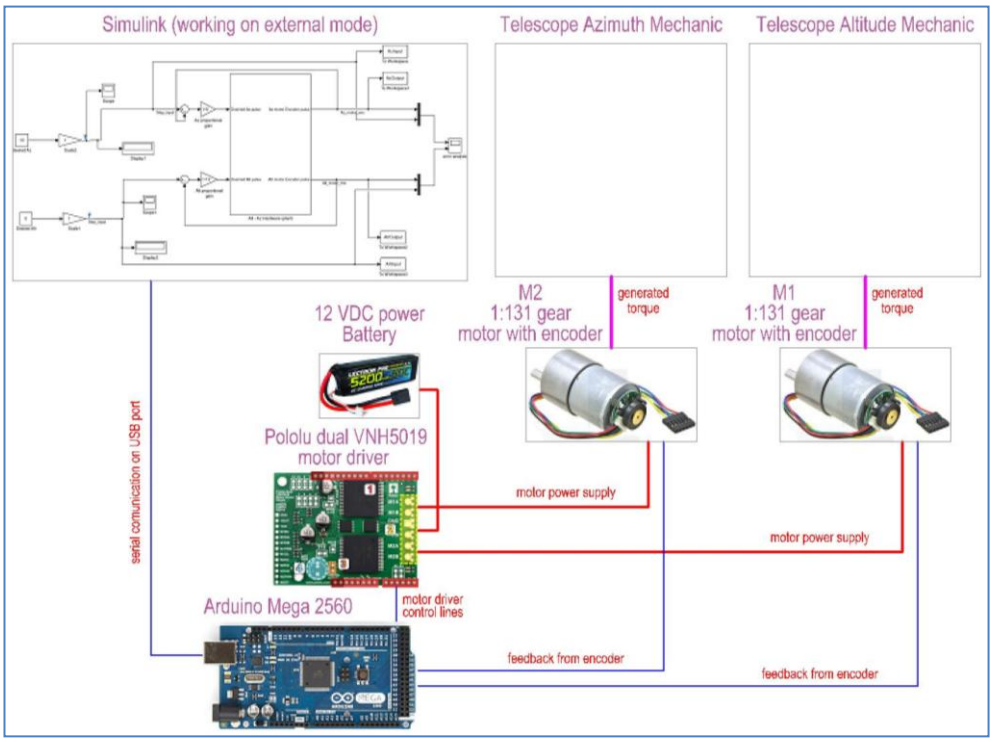


Figure 33: Control System Structure

The control system is designed based on the block diagram shown in Fig.16. For assembling the electronics and software structure, it is preferred to connect *MATLAB Simulink* with the microprocessor *Arduino Mega 2560* (Fig.32). *MATLAB Simulink* supports data-driven control design and provides an environment for Data Acquisition, System Identification, Control Design and Real-time Testing. Simulink software and arduino hardware algorithms were used for design, test and verification stages.

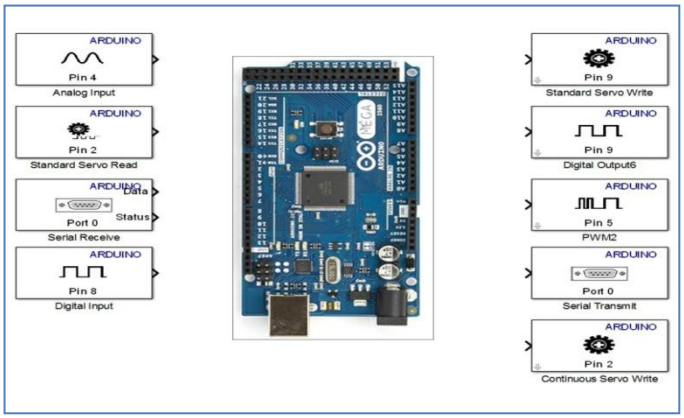


Figure 34: MATLAB Simulink Arduino Package and Arduino Mega 2560

The *Simulink* built-in support for the *Arduino* platform includes Library of *Simulink* blocks that connect to *Arduino* I/O, such as digital input and output, analog input and

output, serial receive and transmit, and servo read-write. Interactive parameter tuning and signal monitoring of applications run on the *Arduino Mega 2560* [19].

5.2.1 Arduino based motor control: (electronic)

We try to explain electronic structures of the telescope mount control. Our physical electronic system is made up of one DC motor coupled for Altitude axis part and one DC motor coupled for Azimuth axis part connected to an Arduino Mega 2560 board via a motor driver (Fig.32). We designed a feedback controller for these motors to track a reference position. The controller generated the appropriate voltage command based on the motor position reference data. When applied to the motor, this voltage causes the motor to generate the torque that turns the motor shaft. We used encoders that are coupled to motor back side to measure the angle of rotation of the motor shaft. The angle measured by the encoder is then fed back to the controller.

The motor driver integrated circuit (IC) increases the current capability and can drive the motor in both directions. We receive the motor position data through serial port on the Arduino board and compute the error between the reference and actual data (the controller input). We send a voltage command (the controller output) to digital output pins on the board as PWM signals. These signals are fed to the driver IC that provides the motor with the appropriate drive currents [19].

The controller must keep the system stable and provide fast reference tracking with minimal (actually zero) steady-state error and overshoot.

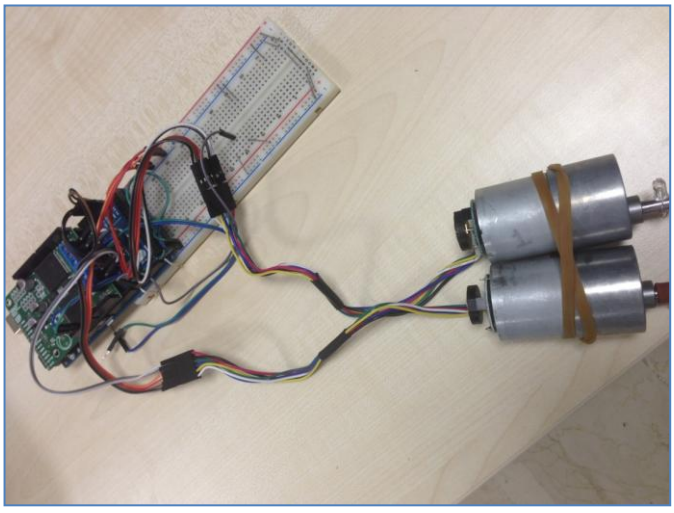


Figure 35: Electronic hardware implementation

5.2.2 Encoders:

Our motor have an integrated quadrature encoder that provides a resolution of 64 counts per revolution of the motor shaft, which corresponds to **8384 counts per**

revolution of the gearbox's output shaft [20]. Shaft encoder is the most common way of providing feed-back to the controller. Shaft encoders have many size and type but they all work with the same principle. Shaft encoder has two main parts; one of them is the encoder disk that is coupled dc motor back-shaft and second U shaped photo-couple.

The encoder disc has 16 holes and two photo-couples. Photo-couple has two sides. One side is IR led sending beams through to encoder disc holes and other side IR sensor receives the beams. IR sensors produce 1-0 signals, during rotation. IR sensors send serial data to controller from Port-A and Port-B in any direction. Our encoders have 16 slits meaning that the encoder produced 16 up and down counts per IR sensor and so as a total of 64 counts per revolution (CPR).

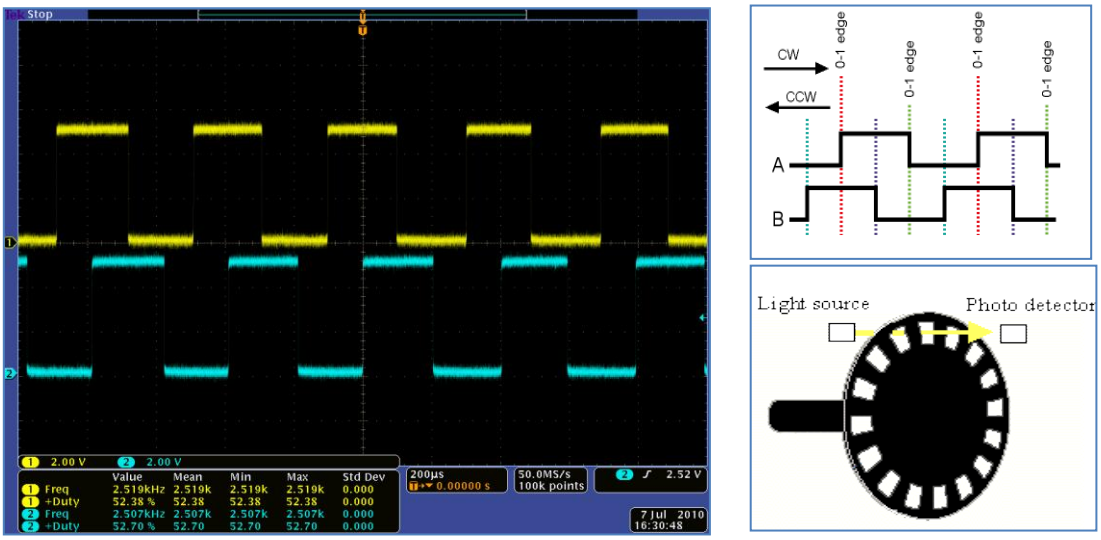


Figure 36: Encoder behavior on oscilloscope screen, Count plots and Slit [20]

We have to calculate total count per telescope mount rotation on 360°.

Encoder count per revolution: 64

DC motor gear ratio: 1:131

Worm wheel gear ratio: 1:34

We need to find resolution on rotational movement. For this reason if we say Total Count per Telescope Revolution (TCPRT) in 360°:

$$TCPTR = 64 * 131 * 34 = 285056\ count \qquad (21)$$

$$TM\ resolution = \frac{360^{\circ}}{285056\ count} * \frac{60'}{1^{\circ}} * \frac{60''}{1'} = 4{,}54\ \frac{arc-sec}{count} \qquad (22)$$

But because of telescope structural errors and electronic hardware limitations we have not been able to reach these resolutions.

Encoder reading and writing process have given a lot of pains. In order to handle the encoder input, initially it was used the standard Digital Read block of Simulink support package for Arduino. It was worked fine when the motor shaft turned manually; the interrupts routine working and the counters were correctly incremented / decremented. However, when the motor powered up the Arduino block froze. The main problem is this; the quadrature encoders are too fast for Arduino meaning that the amount of reading job is too much for Arduino Digital Read block.

One suggested solution is the use of dedicated quadrature decoder chips well known as Quadrature Encoder Interface (QEI) like the LS7266R1.

We decided to read and write procedure from serial ports on Arduino board instead of reading from digital Read function block. We connected encoders to external interrupts pins on the Arduino mega 2560. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. The allowed pins are 2, 3, 18, 19, 20 and 21 on the board.

A Simulink S-Function Block attaches at initialization time, the interrupt service routines, written in C code, as illustrated in appendix C, to the two pins to which the encoder is connected. After that, when the encoder rotates the interrupts service routines update the encoder position. The encoder position is then set as output of the block every time (sample time) the block is executed.

5.3 Real system (plant) model and PID tune:

The Simulink support package has been discussed earlier in Section 5.2. The Real Time System Model has been assembled with extensive usage of this package. The Simulink Model and the sub model figure for plant can be seen at Fig.37 and Fig.38. Feedback from the encoders on both motors is provided by the sfunction (sfunc-enc) block.

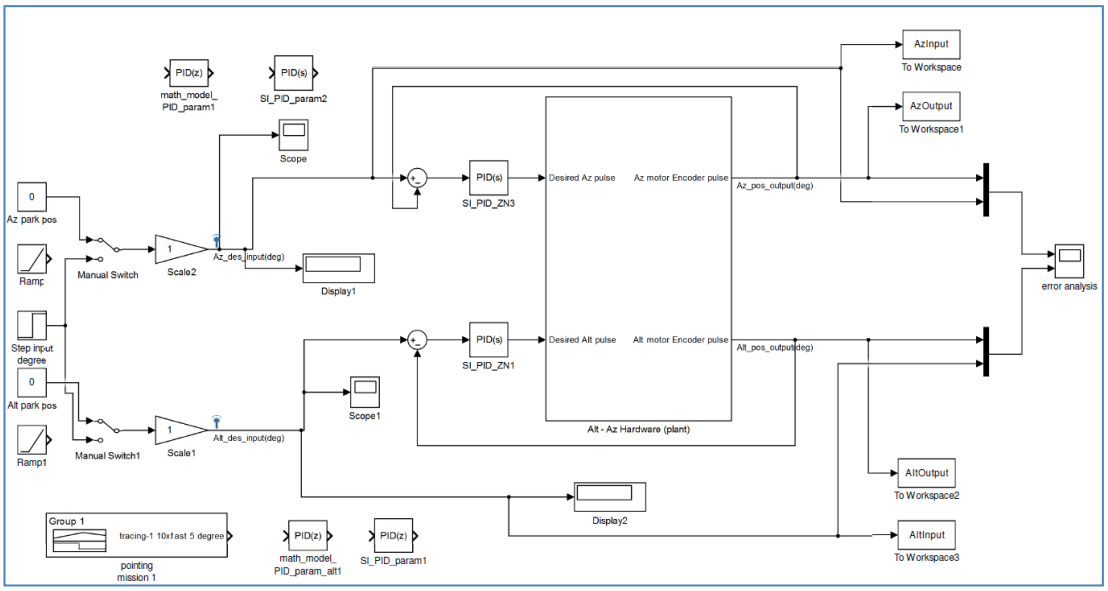


Figure 37: Main control page on Simulink

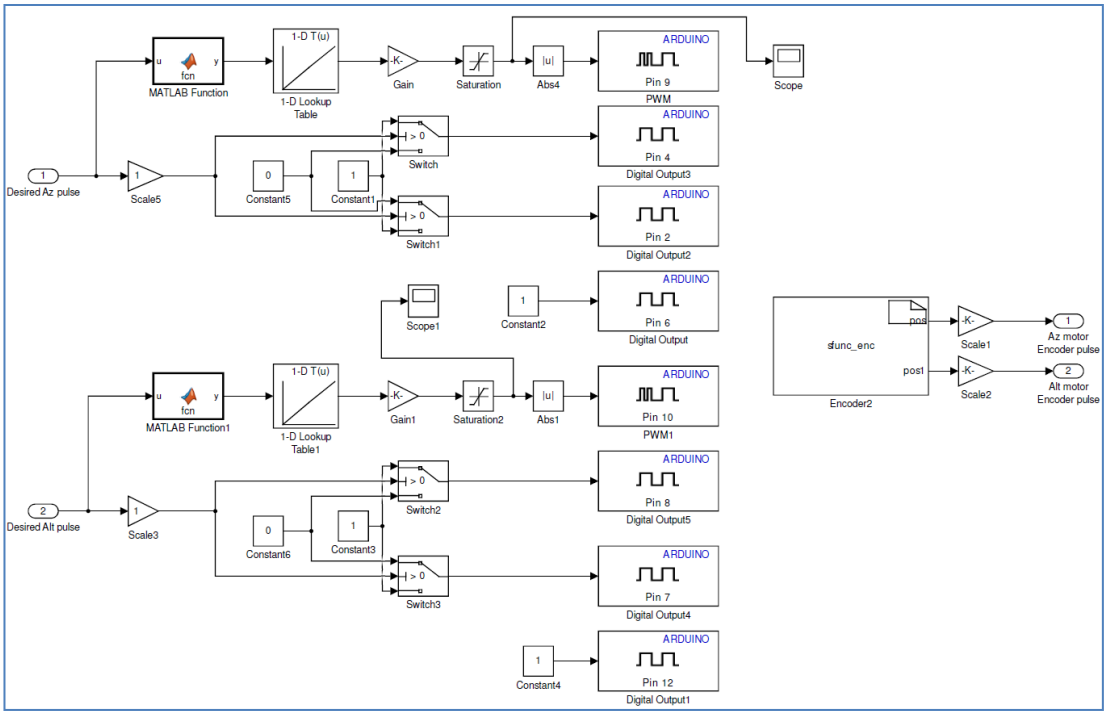


Figure 38: Arduino control page (plant) on Simulink

Comparison operators were used to determine the negative-positive position information at Fig.38 for data input at the direction blocks. Saturation module was used before the PWM module to limit the maximum speed of the axes. Even though the input value is in degrees, value read by the encoders will be in pulses. That's why gain has been implemented to the real model's input and output to execute the conversion.

The system was first run in external mode. The close loop control has been provided by unity feedback. The system has reached to the desired point, yet it show oscillations. 20°, 50°, 0°, -30° were input to the azimuth axis respectively. The result can be seen at Fig.39. The system oscillates within a ±1° degree range.

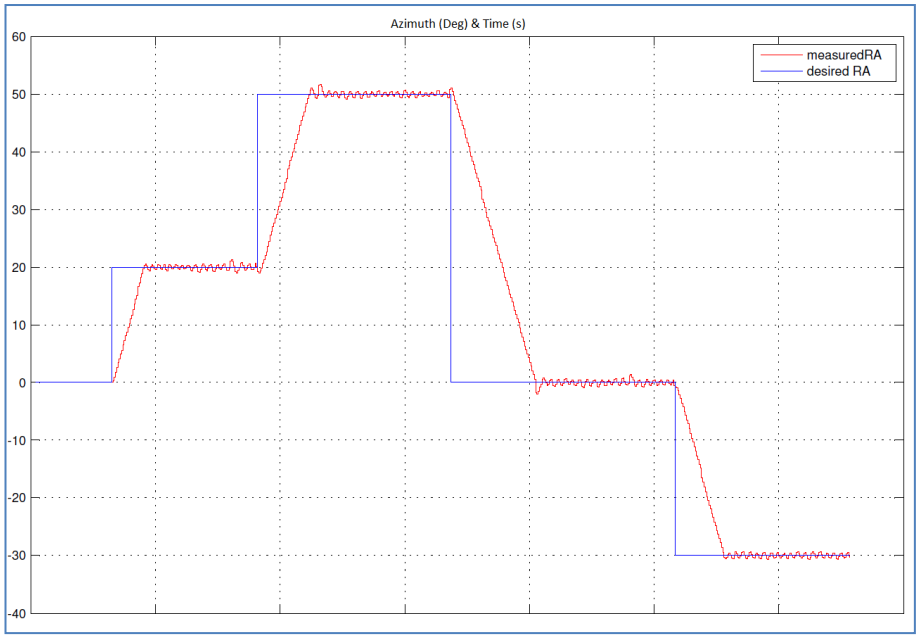


Figure 39: Close Loop Control (CLC) unity feedback 20°, 50°,0°,-30° input response

To eliminate the oscillation, P-controls were added to the azimuth axis and the unity feedback was shutdown at the altitude axis. As seen at Fig.40 and Fig.41, oscillations at the Azimuth axis were eliminated, yet it is still present at the altitude axis.

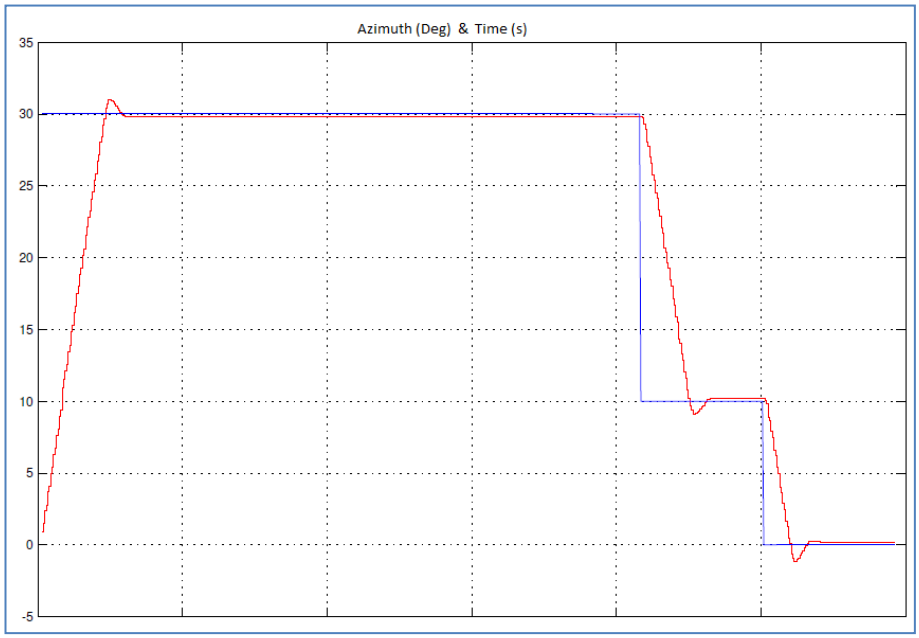


Figure 40: Azimuth axis CLC, P Controller 30°, 10° input response

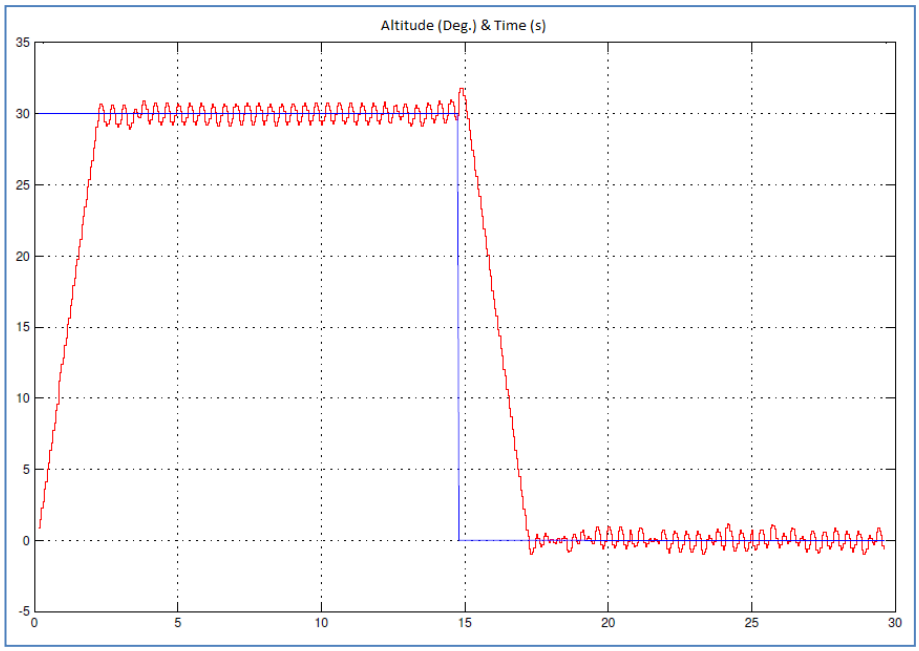


Figure 41: Altitude axis CLC, unity feedback 30°, 0° input response

Later, P-controls applied on both axes. Response of the input for 180° can be seen at Fig.42. An overshoot of 0.1° is observed instead of a 1° degree overshoot.

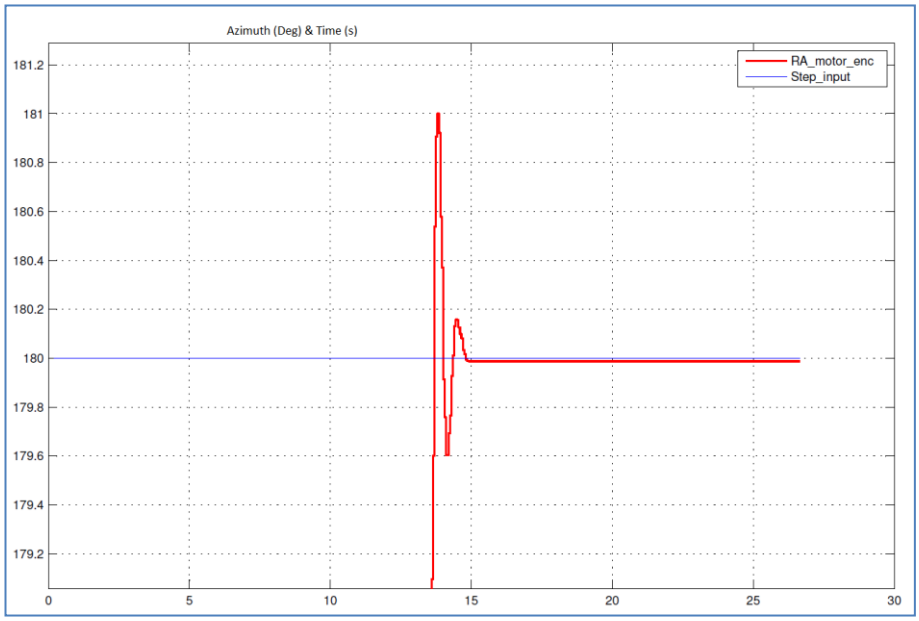


Figure 42: Azimuth axis CLC, P Controller 180° input response

After the initial results, the coefficients obtained from the mathematical model (Fig.25, Fig.26.a.b.c and Fig.27.a.b.c) of P and PID was used on the real system model. Results are provided at Fig.43 and Fig.44.

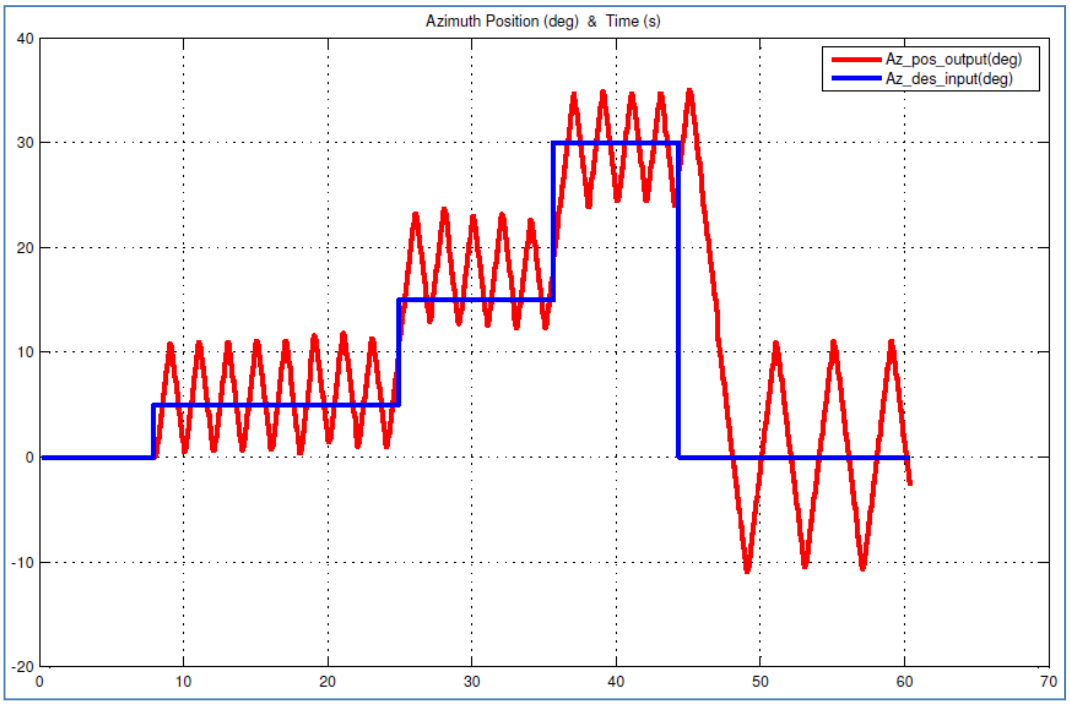


Figure 43: Azimuth axis CLC, P Controller 5°, 15°, and 30° input response

P-controls respond to the system as seen at Fig.43 yet an oscillation of ±5° and a steady state error of 2° is present. PID controls provided at Fig.44 did not respond to the system.

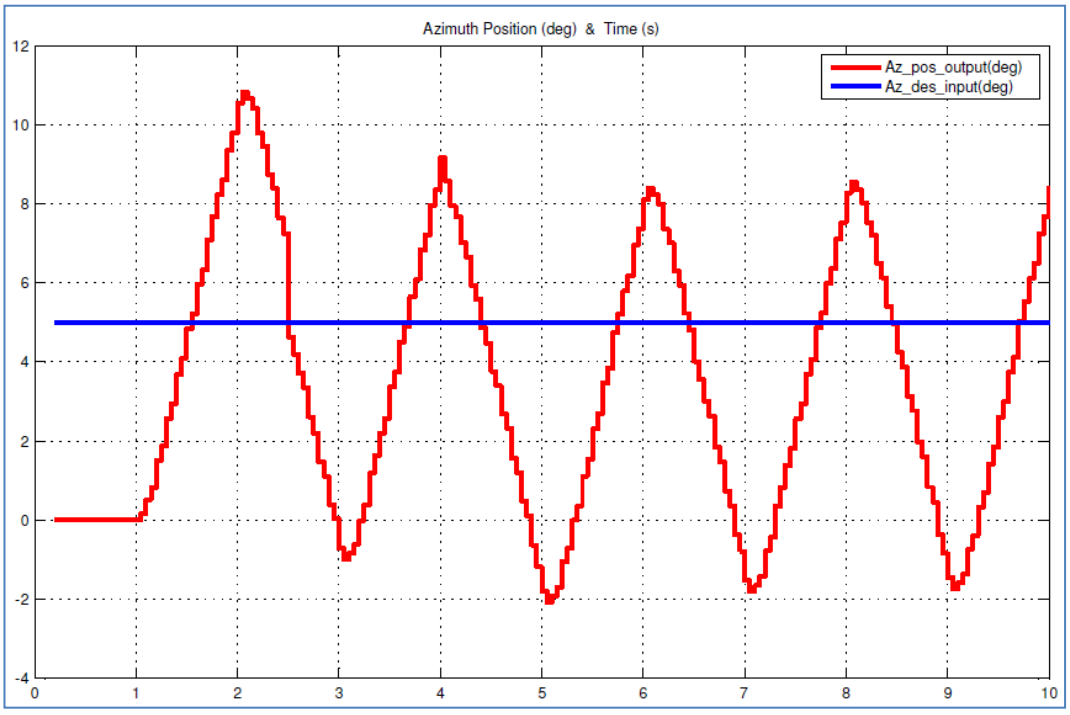


Figure 44: Azimuth axis CLC, PID Controller 5° input response

After these tests, it has been decided to determine the PID parameters using system identification method.

5.3.1 System Identification:

System identification is the art of building mathematical models of dynamical systems using experimental data. It is an iterative procedure. A real system is often very complex. A model is merely a good approximation. Procedure: (a) Collect data, (b) Choose Model Structure, (c) Determine the best model within a structure, (d) Model validation [21].

We are using system identification toolbox in MATLAB Simulink environment. As above explanation we collected input and output data from our plant when it was working real-time mode. The data set used is shown in Fig.45.

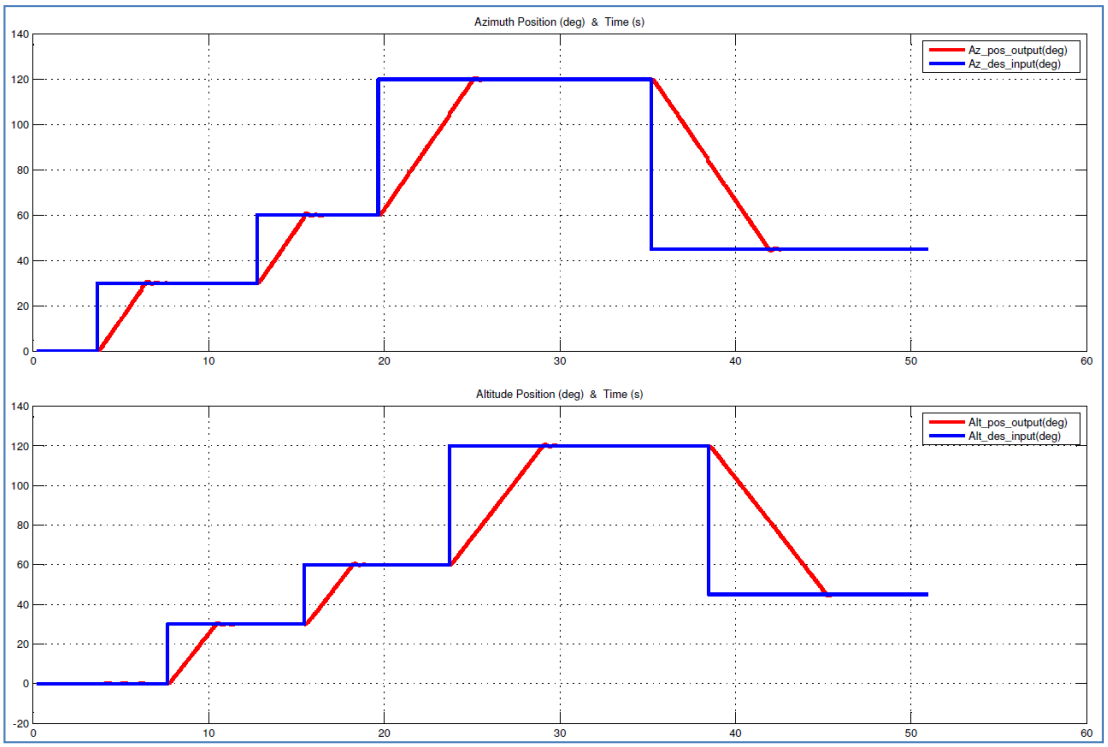


Figure 45: Altitude & Azimuth axis P Controller 30°, 60°, 120°, 45° input response

We give 30°, 60°, 120°, 45°input and took output produced by P controller feedback. After that we followed the procedure and put to input and output data to Matlab system identification toolbox. This toolbox and a part of procure are shown in Fig.46.

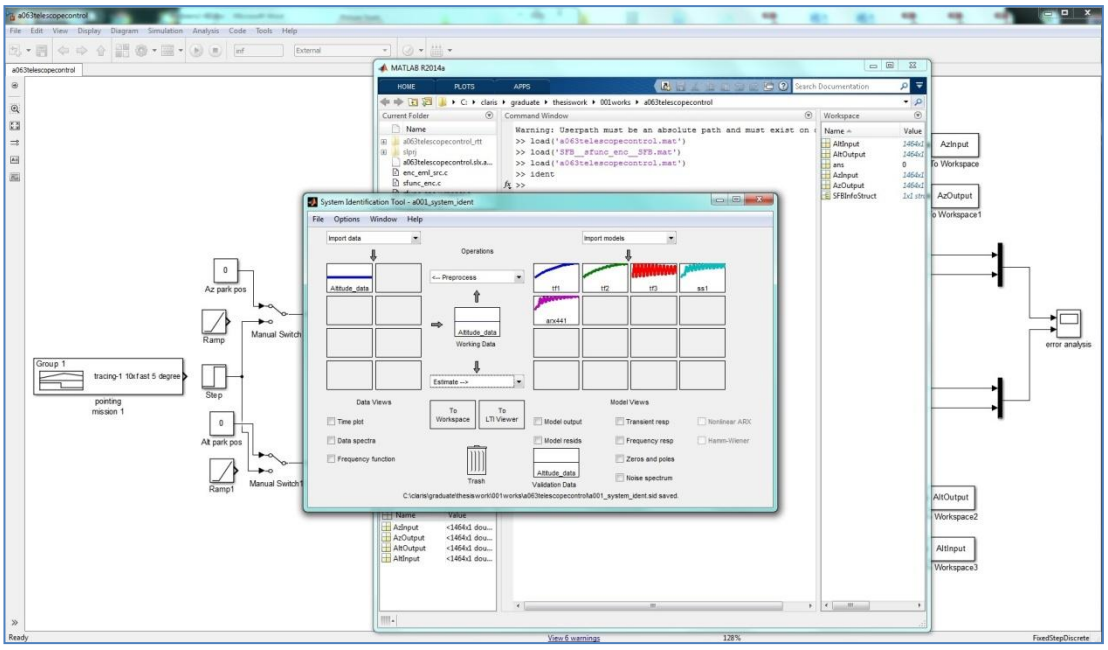


Figure 46: Matlab System Identification Toolbox and part of procedure

When preparing data for identifying models, you need to specify information such as input-output channel names, sampling time, and inter sample behavior. The toolbox lets you attach this information to the data, which facilitates visualization of data, domain conversion, and various preprocessing tasks. The toolbox estimates the impulse and frequency responses of the system directly from measured data [22].

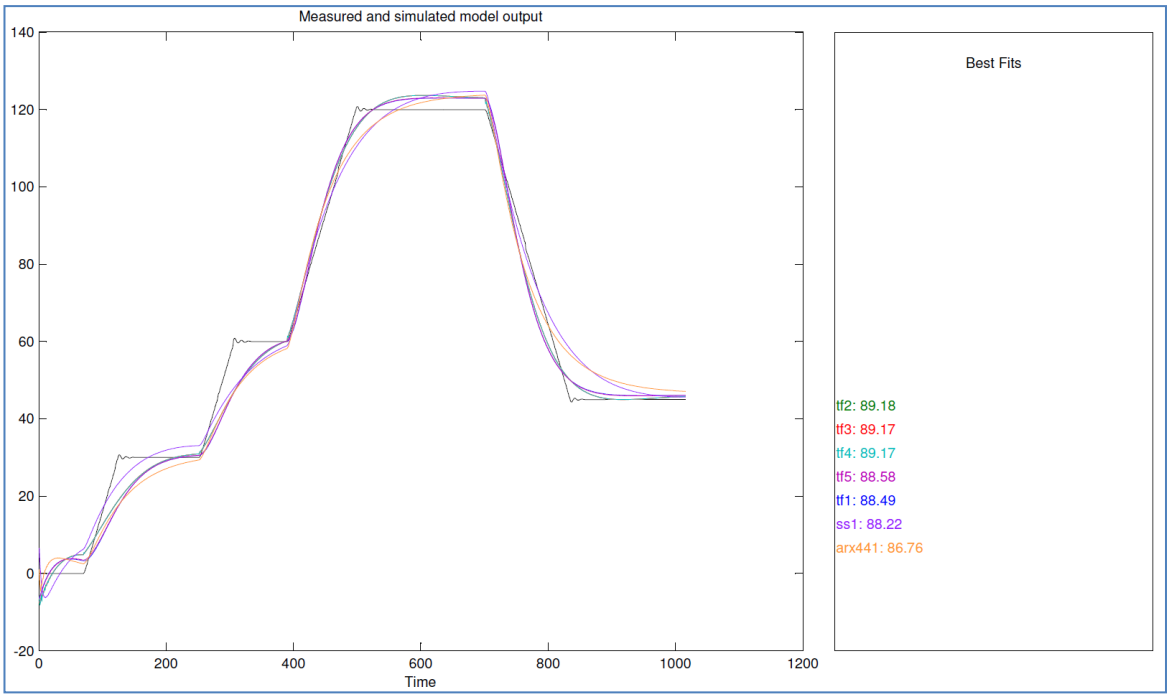


Figure 47: Measured data and estimated multiple models

Fig.47 shows us measured data and estimated multiple models with together represent percentage of the real system. As you see the best option is 2 poles, 2 zeros transfer function (tf2) which is % 89.18.

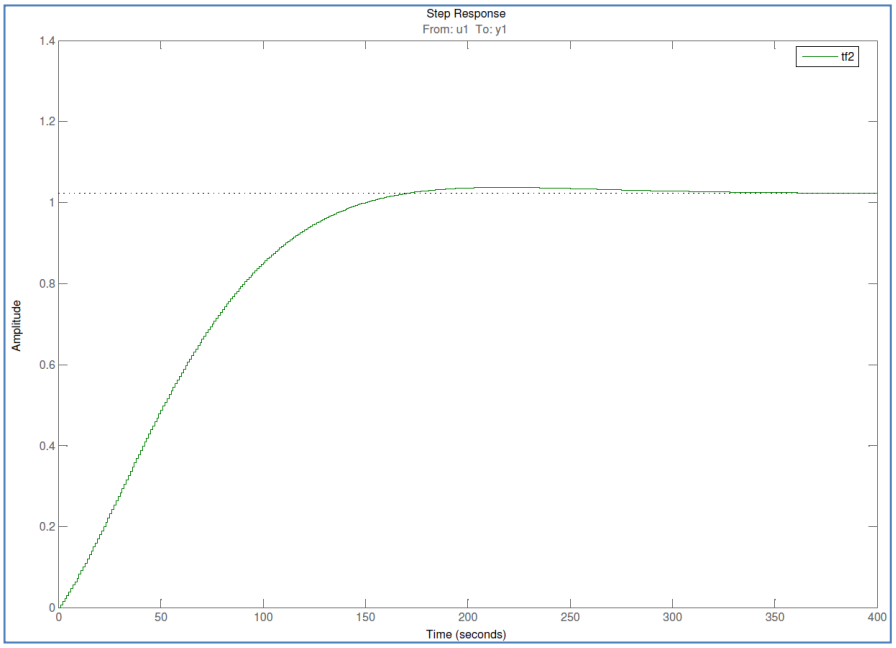


Figure 48: Step Response of Transfer Function 2 (Tf2)

And system identification tool gave us step response for our best chose tf2. (Fig.48)

$$tf2 = \frac{0.0072z^{-1} - 0.0066z^{-2}}{1 - 1.9630z^{-1} - 0.9635z^{-2}} \qquad (23)$$

Table-6 properties of tf2 transfer function

Name: tf2 Discrete-time identified transfer function. Sample time: 1 seconds
Number of poles: 2 Number of zeros: 2 Number of free coefficients: 4
Estimated using TFEST on time domain data “Azimuth data".
Fit to estimation data: 88.53% (simulation focus)

We used this transfer function for finding PID parameters. Fig.49 shows both axis system models for PID tunes.

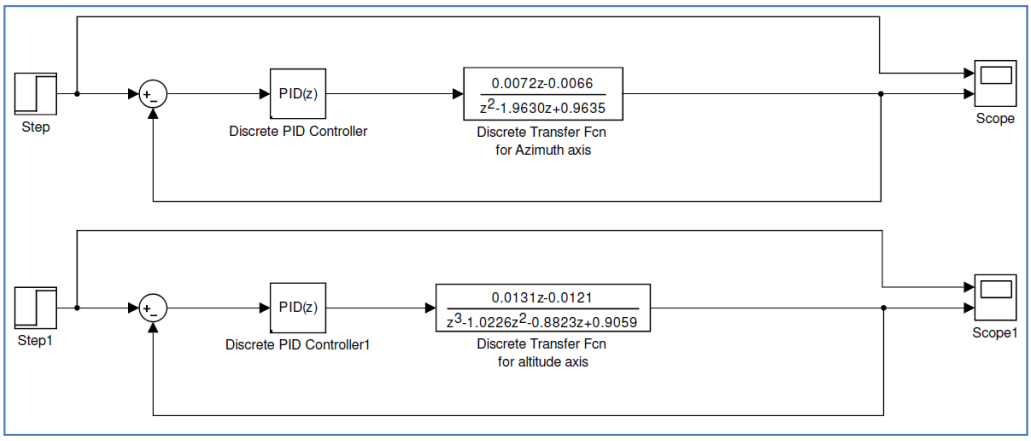


Figure 49: SI Based system control model

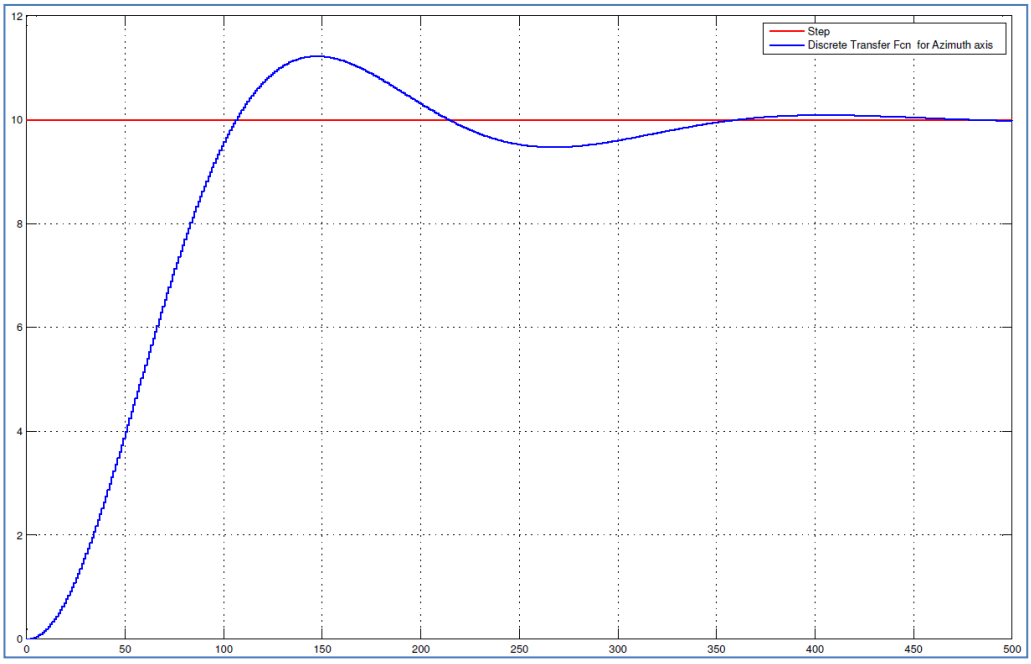


Figure 50: SI Based system control model step response

Fig.50 is showing us the result of the PID tune parameters. Now we have to move these PID parameters to actual systems.

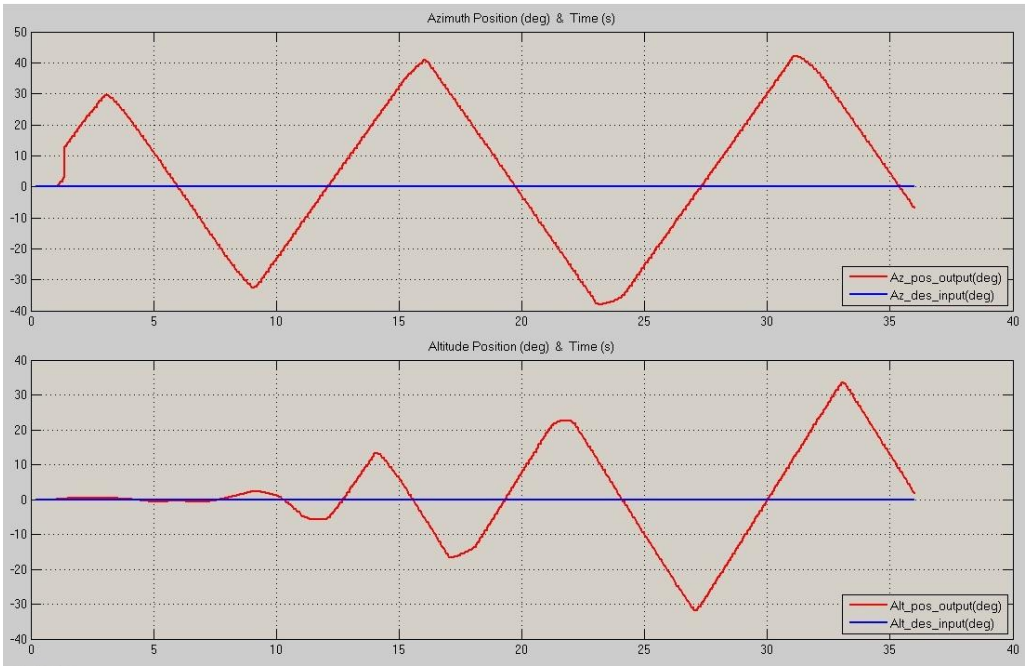


Figure 51: Actual system response for both axes for SI based PID Controller

When we applied to PID parameters to actual systems they were unstable (Fig.51). Unfortunately we have to pass to manual tuning method.

5.3.2 Manual Tuning:

If the system must remain online, one tuning method is to first set K_i and K_d values to zero. We should increase the K_p until the output of the loop oscillates, then the K_p should be set to approximately half of that value for a "quarter amplitude decay" type response. Then we should increase K_i until any offset is corrected in sufficient time for the process. However, if the value of K_i is increased too much, then it will cause instability. Finally, we should increase K_d, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, if we increase K_d too much, it will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the set point more quickly; however, some systems cannot accept overshoot, in which case an over-damped closed-loop system is required, which will require a K_p setting significantly less than half that of the K_p setting that was causing oscillation [23].

We follow the manual tuning procedure and took the pest result with PI controller. Finding P=0.155 and I=0.012 for azimuth axis and P=0.160 and I=0.018 for altitude axis. Results are shown in Fig.52.

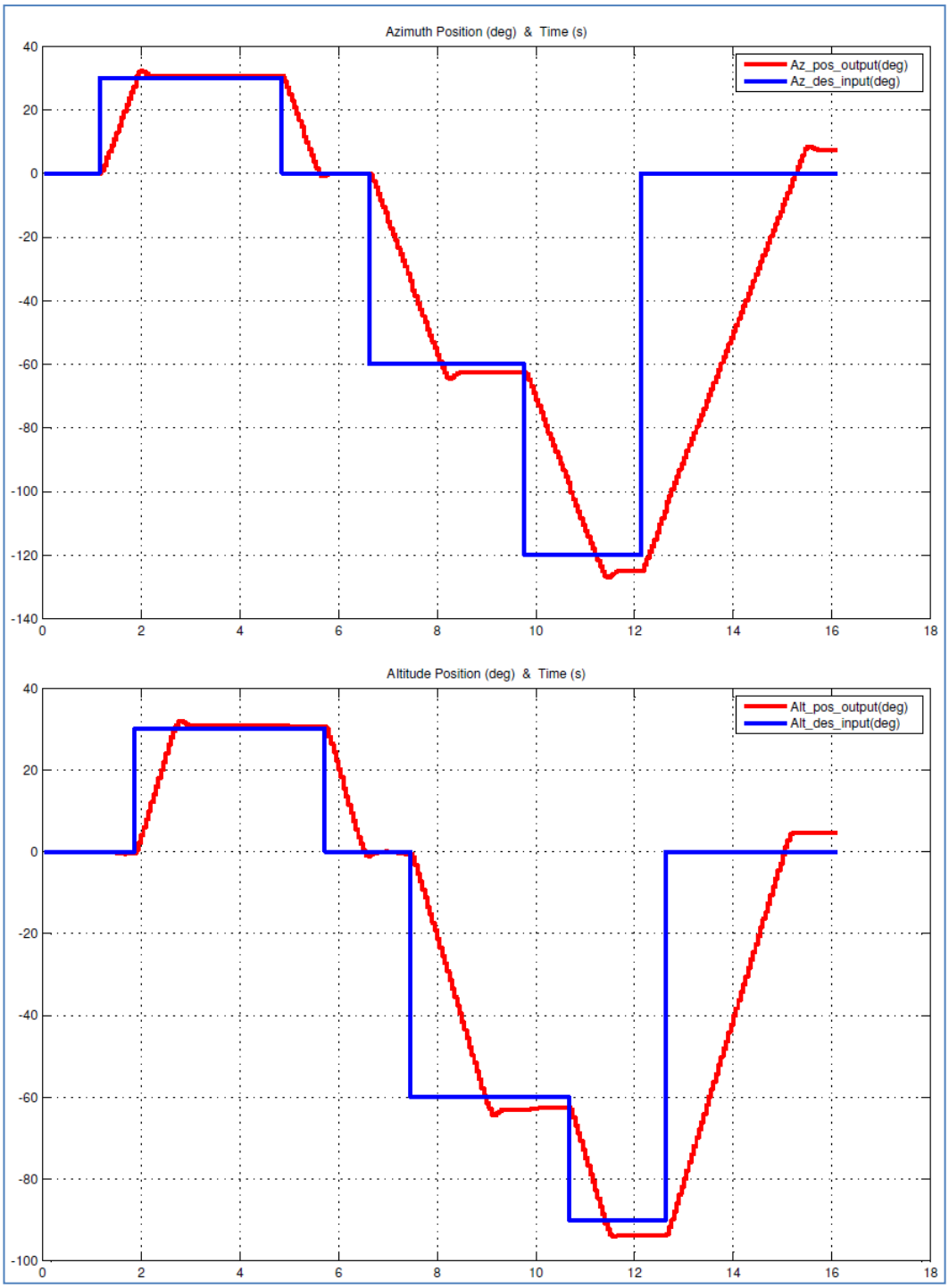


Figure 52: Actual system response for both axes for Manual Tune based PI controller

These plots are saying us azimuth axis steady state error (pointing error) ±0.12° and Overshoot ±1.3°. Likewise altitude axis steady state error (pointing error) ±0.22° and overshoot ±1.2°.

We are going to examine one more figure and will finish this part. This figure (Fig.53) represents tracing performance of the both axis. We set a ramp input for

varying degree in time. World surface angular velocity is 1/240 (°/s). We simulate this displacement input as 10 times faster in half time range, meaning that 5/120 (°/s) measurement results are shown in this Fig.53.

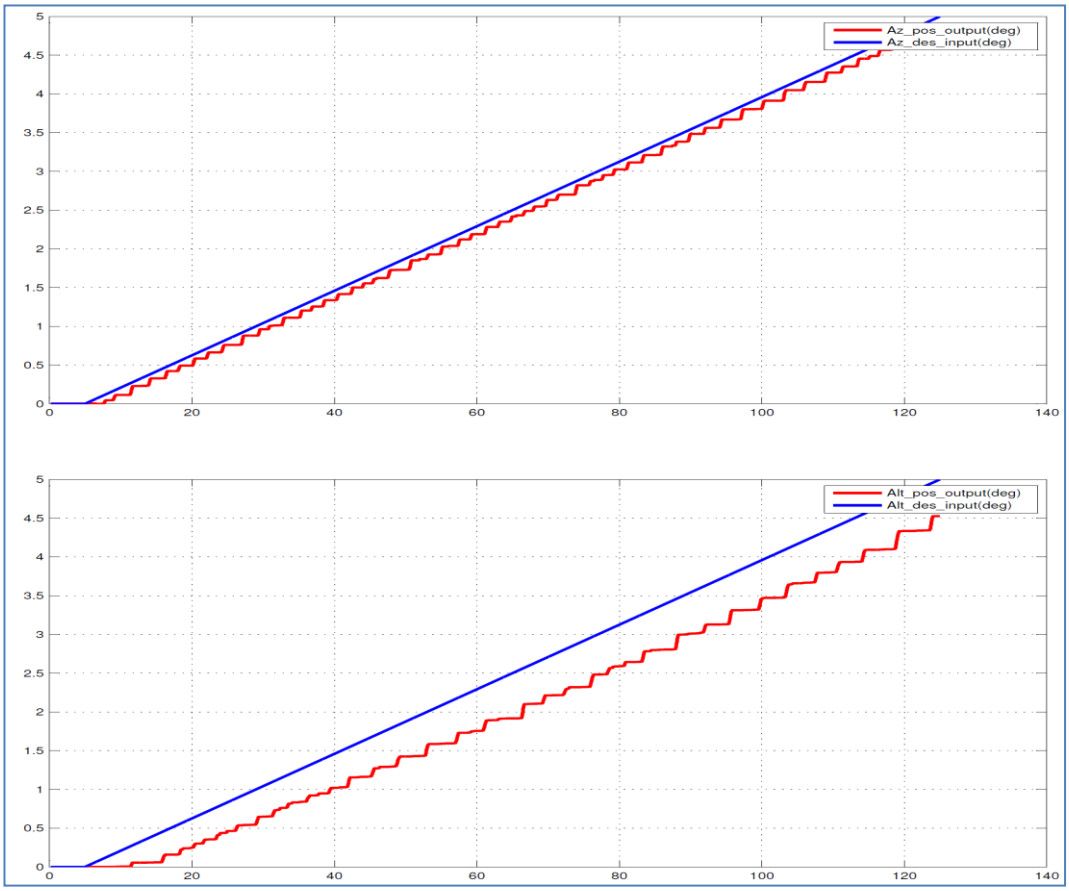


Figure 53: Actual system ramp input response for both axes for PI Controller

From this figures, we see that the azimuth axis doesn't have tracking error but it has tracking precision error. Altitude axis has both tracking error and tracking precision error.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions:

According to the aim of this thesis, we developed two degrees of freedom fork mount actuating mechanism. This work covers both design and manufacture of azimuth and altitude controls of the small class optical telescopes. The system is a Cassegrain type telescope. It has an OTA designed and manufactured by the Aselsan Company and a fork mount designed in this thesis. We used a well known PID controller to control the positions of both altitude and azimuth axis of the fork mount assembly. During the experiments of the controller, we used PI control instead of PID since the requirements to operate the telescope. The needs for the telescope are to have a zero steady-state error with small overshoot. Since the speed of the rotations are so small that there is no need to use D-control. The experiments showed that the PI control gives very reasonable results. We can achieve zero steady-state error with a good system response. The transient response is in the design range as well. The system was implemented in Matlab/Simulink environment with Arduino Mega tools. They facilitated the implementation procedures. Unfortunately we couldn't reach planning precision and accuracy because of the mechanical structure that we used in the system. The problem is that the worm gear that we used has a bad resolution worm gear and also produced backlash effect. This is a type of non-linearity affecting the behavior of the systems. However, since we are trying to prove the concept of control and achieve to manufacture a working prototype of telescope, the result is accepted by us and encouraged us to produce a professional working telescope using the developed hardware and software in this thesis.

Last chapter we put a lot of results related with real system control. These results were obtained from the shafts of the dc motors and these not cover mechanical errors available on the OTA. Our Alt-Az telescope mount could turn towards desired position but its control system has to convert horizontal coordinate to equatorial for tracing of stars or out of earth objects.

Figure 54: Completed telescope mount prototype

6.2 Future works:

The telescope prototype system that is manufactured during the Works of this thesis needs to be improved to become an operational telescope. The TM part is fully controlled except the problem of backlash. The components of the OTA are available but they need to be integrated in a way such that the optics will focus properly. Every component of the optical setup (OTA) needs to be fully working in order to convert the mount into an operational telescope. The OTA currently consists of a parabolic primary mirror (M1) with a radius of 295mm and a hyperbolic secondary mirror (M2) with a radius of 57,7mm. The surface curvatures of the mirrors, focal lengths and the distance between M1-M2 have all been calculated using Zemax. The monolithic telescopic tube is made of delrin. The optical tube, M1 and M2 mirrors are kindly provided by Aselsan for this study. It is necessary to assemble these components. To get a proper and precise focus, the mirrors should be aligned in three axes, and the distance between M1-M2 mirrors should be adjusted according to the calculated value of the Zemax output, which needs extensive usage of an optical

laboratory. A set of eye-pieces with different focal lengths will be necessary to see the optical image after setting the focus. In order to get a photometric or digital recording of the image, a digital camera or CCD will be necessary.
To use the telescope in an observatory for scientific purposes in a professional manner, devices such as a filter wheel, focuser, collimator will be required on the focal plane. The telescope control software / graphical user interface needs to be ASCOM platform compliant in order to control it with current astronomical software. If a professional usage is not necessary, the telescope can be used for popular events with its basic features.

The telescopes pointing and tracking mechanism capabilities have been explained within this thesis. With further studies in an optical laboratory, tracking and observing objects such as satellites will be available. This will only be possible with feedback from the image processing software to the telescope control system.

The telescope, even it is a prototype, is the first national telescope developed so far. We hope that it will encourage both scientific world and commercial world to design and manufacture professional telescopes. We again hope that we will get a proper place among the other nations in this field.

REFERENCES

[1] http://www.davidreneke.com/buying-and-using-a-telescope/, Visited Date; 02.10.2014

[2] http://www.astronomy2009.org/news/updates/490/, Visited Date; 07.10.2014

[3] http://www.atmturk.org, Visited Date; 08.10.2014

[4] http://www.astrophotography.id.au/; Visited Date; 18.11.2014

[5] http://www.telescopes.com/blogs/helpful-information/18966596-understanding- telescopes; Visited Date; 18.11.2014

[6] Sinclair, S. An Autonomous Telescope Implementation at the USQ Mount Kent Observatory, University of Southern Queensland Faculty of Sciences, Master of Science, February, 2010.

[7] http://www.united-optics.com/ support/mechanical_term_and_characteristic_of_telescope/ mechinanical_term_and_characteristic_of_telescope.html; Visited Date; 05.12.2014

[8] http://www.vikdhillon.staff.shef.ac.uk/teaching/phy217/ telescopes/phy217_tel_equatorial.html, Visited Date; 23.12.2014

[9] https://www.pololu.com/product/1995, Visited Date; 09.01.2015

[10] http://machinedesign.com/technologies/methods-minimize-gear-backlash, Visited Date; 28.01.2015

[11] http://www.skf.com/group/products/bearings-units-housings/ball-bearings/deep-groove-ball-bearings, Visited Date; 17.03.2015

[12] Gawronski, Wodek. Modeling and Control of Antennas and Telescopes, Springer. 2008

[13] Richard C. Dorf and Robert H. Bishop, Modern Control Systems, Prentice Hall, 2001.

[14] Nise, Norman S. Control System Engineering, John Wiley & Sons, INC. 2008.

[15] Ogata, Katsuhiko, Model Control Engineering, 1997

[16] Cadzow, J.A. (1973). Discrete-Time Systems, Englewood Cliffs, NJ, USA.

[17] https://www.arduino.cc/en/Main/ArduinoBoardMega2560, Visited Date; 10.03.2015

[18] https://www.pololu.com/docs/0J49/all, Visited Date; 19.04.2015

[19] http://www.mathworks.com/hardware-support/arduino-simulink.html, Visited Date; 07.02.2015

[20] http://www.robotgear.com.au/Product.aspx/Details/489-100-1-Metal-Gearmotor-37Dx52L-mm-with-64-CPR-Encoder, Visited Date; 28.04.2015

[21] Pelckmans, K. System Identification course notes, Uppsala University, Information Technology. 2010

[22] http://www.mathworks.com/products/sysid/features.html, Visited Date; 12.04.2015

[23] Kurien, Manju at all, Overview of Different Approaches of PID Controller Tuning, E-Issn:2321-9637, January 2014

[24] http://www.globalspec.com/learnmore/motion_controls/power_transmission/gears/worms_worm_gears, Visited Date; 02.05.2015

APPENDIX A

DATASHEETS OF EQUIPMENTS

A.1 Arduino Mega 2560:

Arduino MEGA 2560

CE

Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

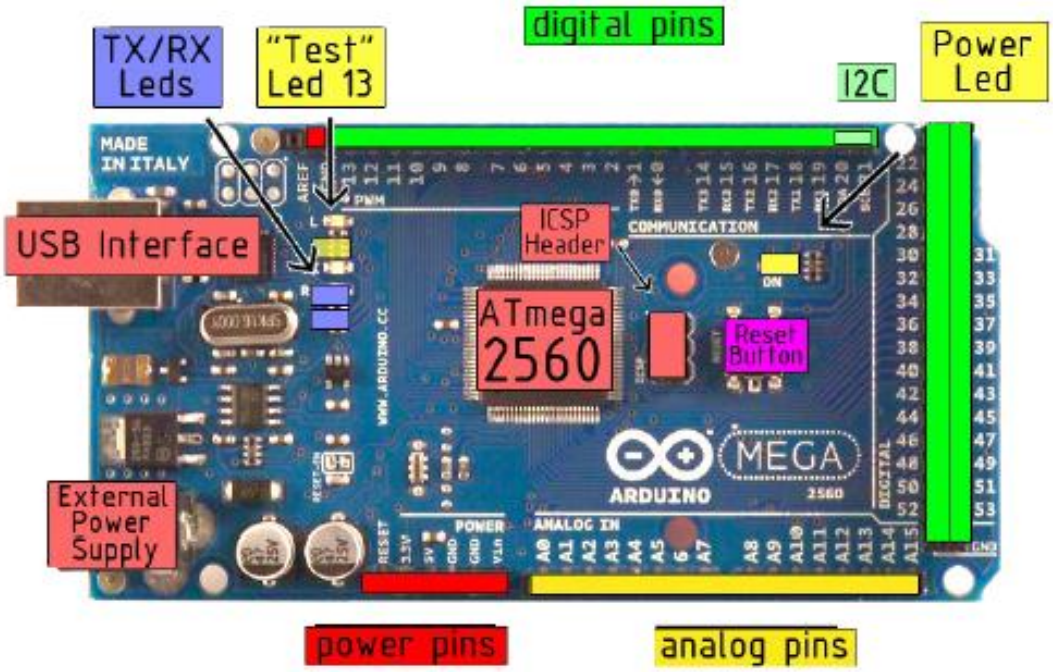
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

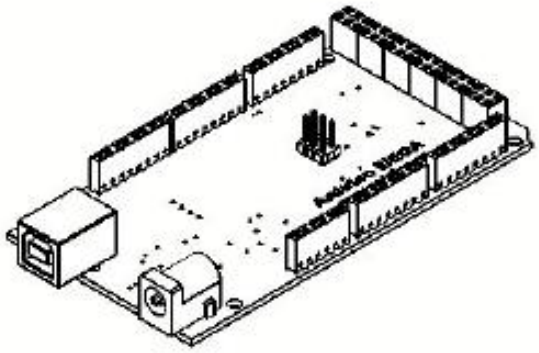
Summary

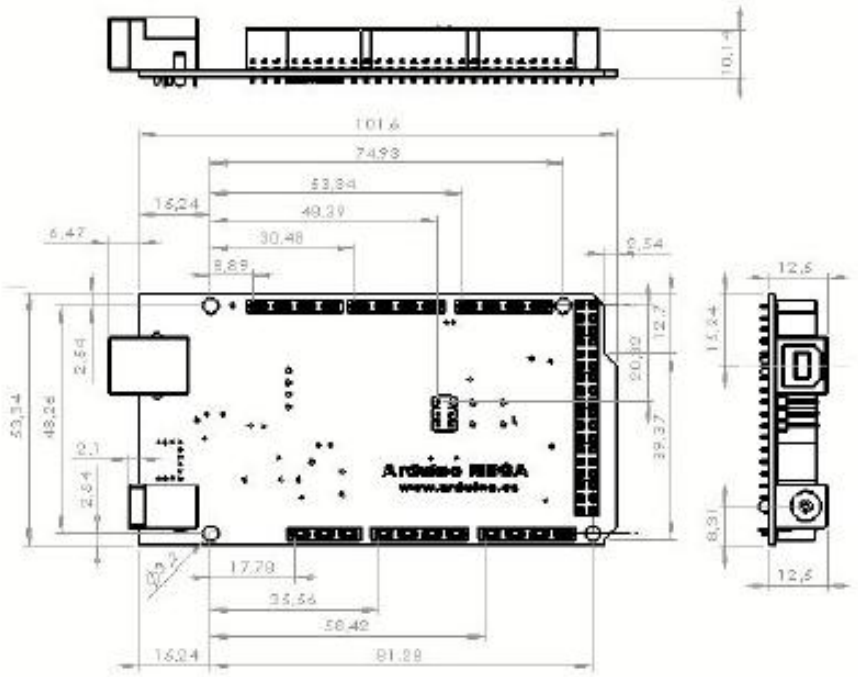
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

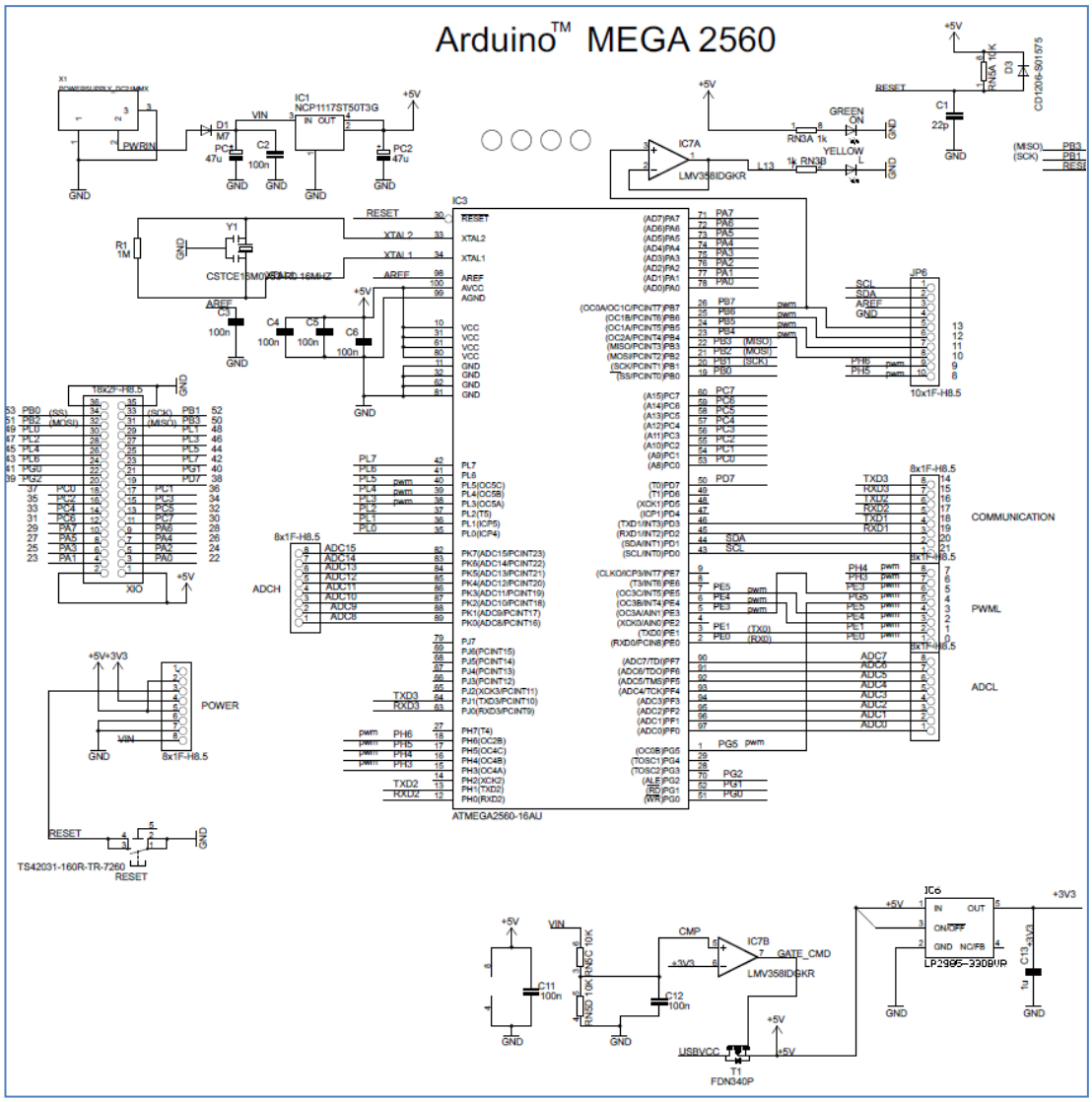
the board



Dimensioned Drawing






Arduino™ MEGA 2560
ATMEGA2560-16AU
IC1
NCP1117ST50T3G
IC7A
LMV358IDGKR
IC7B
LMV358IDGKR
IC6
LP2985-33DBVR
T1
FDN340P
TS42031-160R-TR-7260
RESET
POWER
COMMUNICATION
PWML
ADCL
ADCH
CSTCE16M0V53-R0
16MHZ
GREEN
YELLOW
CMP
GATE_CMD
USBVCC

A.2 Pololu Dual VNH5019 Motor Driver Shield:

Pololu Dual VNH5019 Motor Driver Shield User's Guide

1. Overview

This user's guide focuses on the **latest version (ash02b)** **[http://www.pololu.com/product/2507]** of the Pololu dual VNH5019 motor driver shield, but most of the information also applies to the earlier **ash02a** **[http://www.pololu.com/product/2502]** version. See **Section 8** for details about the differences between board revisions.

Pololu dual VNH5019 motor driver shield for Arduino.

The **Pololu dual VNH5019 motor driver shield for Arduino** **[http://www.pololu.com/product/2507]** and its corresponding Arduino library make it easy to control two bidirectional, high-power DC motors with an **Arduino** **[http://www.pololu.com/product/2191]** or compatible board, such as the **A-Star 32U4 Prime** **[http://www.pololu.com/category/165/a-star-32u4-prime]**. The board features a pair of robust VNH5019 motor drivers from ST, which operate from 5.5 to 24 V and can deliver a continuous 12 A (30 A peak) per channel, and incorporates most of the components of the typical application diagram on page 14 of the **VNH5019 datasheet** **[http://www.pololu.com/file/download/VNH5019A-E.pdf?file_id=0J504]** (629k pdf), including pull-up and protection resistors and FETs for reverse battery protection. It ships fully populated with its SMD components, including the two VNH5019 ICs, as shown in the picture to the right; stackable Arduino headers and terminal blocks for connecting motors and motor power are included but are not soldered in.

This versatile motor driver is intended for a wide range of users, from beginners who just want a plug-and-play motor control solution for their Arduinos (and are okay with a little soldering) to experts who want to directly interface with ST's great motor driver ICs. The Arduino pin mappings can all be customized if the defaults are not convenient, and the VNH5019 control lines are broken out along the left side of the board, providing a convenient interface point for other microcontroller boards. This versatility, along with an option to power the Arduino directly from the shield, sets this board apart from similar competing motor shields.

3. Getting Started with an Arduino

As with virtually all other Arduino shields, connections between the Arduino and the motor driver are made via extended stackable headers that must be soldered to the through-holes along the top and bottom edges of the shield. This section explains how to use this motor driver as an Arduino shield to quickly and easily add control of up to two DC motors to your Arduino project. For information on how to use this board as a general-purpose motor driver controlled by something other than an Arduino, see **Section 4**.

3.a. What You Will Need

The following tools and components are required for getting started using this motor driver as an Arduino shield:

- **An Arduino or compatible control board.** Using this product as an Arduino shield (rather than a general-purpose motor driver board) requires an **Arduino [http://www.pololu.com/product/2191]**. This shield should work with all Arduinos and Arduino clones that behave like a standard Arduino. You will also need a USB cable for connecting your Arduino to a computer. We have specifically tested this shield (using our Arduino library) with:
  - **A-Star 32U4 Prime [http://www.pololu.com/category/165/a-star-32u4-prime]**
  - **Arduino Uno [http://www.pololu.com/product/2191]** (both original and R3)
  - **Arduino Leonardo [http://www.pololu.com/product/2192]**
  - **Arduino Due [http://www.pololu.com/product/2193]***
  - **Arduino Mega 2560 [http://www.pololu.com/product/1699]**
  - Arduino Duemilanove (both with ATmega168 and ATmega328P)
  - chipKIT Max32 Arduino-Compatible Prototyping Platform (PIC32-based Arduino clone)
- **A soldering iron and solder.** The through-hole parts included with the shield must be soldered in before you can plug the shield into an Arduino or before you can connect power and motors. An **inexpensive soldering iron [http://www.pololu.com/product/156]** will work, but you might consider investing in a higher-performance, adjustable soldering iron if you will be doing a lot of work with electronics.
- **A power supply.** You will need a power supply, such as a battery pack, capable of delivering the current your motors will draw. See the *Power Connections and Considerations* portion of **Section 3.c** for more information on selecting an appropriate power supply.
- **One or two brushed DC motors.** This shield is a dual motor driver, so it can independently control two bidirectional brushed DC motors. See the *Motor Connections and Considerations* portion of **Section 3.c** for more information on selecting appropriate motors.

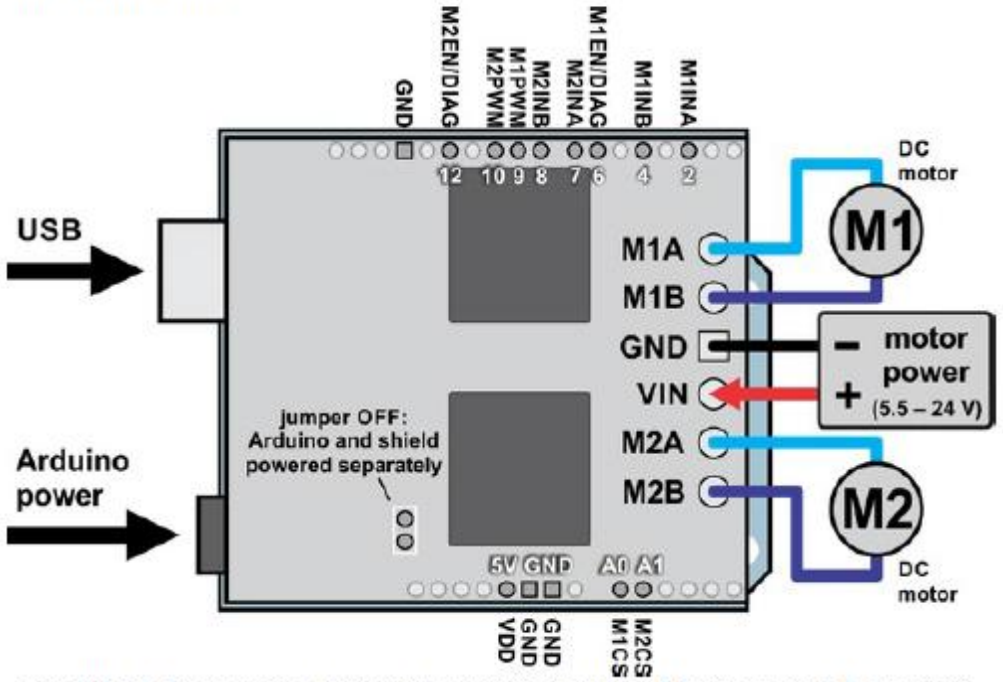
> * **Note for Due users:** The voltage on the current sense pins will exceed the Due's 3.3 V limit when the current draw exceeds ~23 A. The Due should generally be able to handle this since the MCU's integrated protection diodes will clamp the input voltage to a safe value (and since the CS circuit has a 10 kΩ resistor in series with the output, only a few hundred microamps at most will flow through that diode). However, if you really want to be safe, you can use a 3.3 V zener diode to clamp the current sense output voltage to a maximum of ~3.3 V. If you want to get the full range of current feedback while using the Due, you can disconnect the shield's current sense pins from the Due and then reconnect them through a voltage divider; see **Section 6.a** for more information.

3.b. Assembly for Use as an Arduino Shield

1. **Stackable Arduino headers:** Before you can use this board as an Arduino shield, you need to solder four of the five included Arduino header strips to the set of holes highlighted in red in the picture above. The headers should be oriented so that the female sockets rest on the top side of the shield and face up while the male pins protrude down through the board, and the solder connections should be made on the underside of the shield. The newest Arduino boards, including the Uno R3 and the Leonardo, use one 10×1 header, two 8×1 headers, and one 6×1 header, as shown in the left picture below; older Arduino boards use two 8×1 headers and two 6×1 headers, as shown in the right picture below (the two pairs of pins highlighted in darker red should not be populated if you are using this board with an older Arduino that does not support these additional pins). Please make sure you solder the appropriate headers for your particular Arduino!

2. **Motor and power connections:** The six large holes/twelve small holes on the right side of the board, highlighted in yellow in the above diagram, are the motor outputs and power inputs. You can optionally solder the included 5mm-pitch terminal blocks to the six large holes to enable temporary motor and motor power connections, or you can break off a 12×1 section of the included 0.1" header strip and solder it into the smaller through-holes that border the six large motor and motor power pads. Note, however, that the terminal blocks are only rated for 16 A, and each header pin pair is only rated for a combined 6 A, so for higher-current applications,

3.c. Shield Connections



Dual VNH5019 motor driver shield with an Arduino (shield and Arduino powered separately).

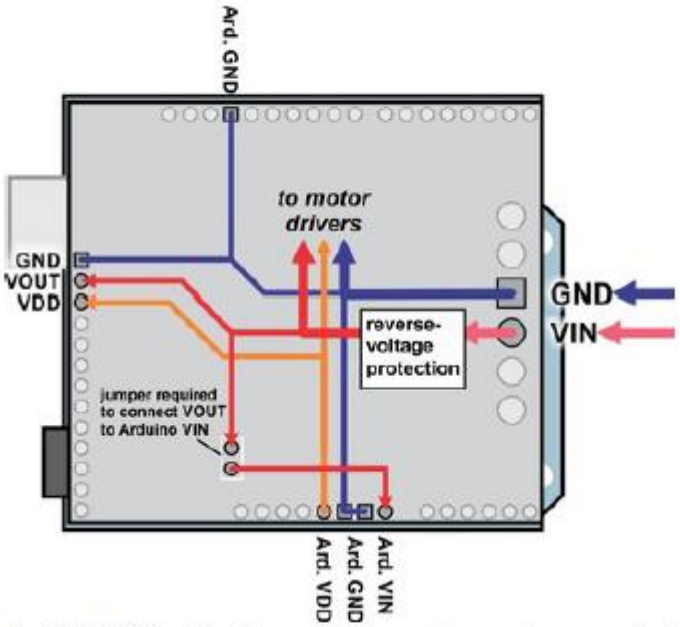
All of the necessary logic connections between the Arduino and the motor driver shield are made automatically when the shield is plugged into the Arduino. However, the shield's power connections must be made directly to the shield itself via its large VIN and GND pads. The picture above shows the typical connections involved in using this board as an Arduino shield.

Default Arduino Pin Mappings

The following table shows how the shield connects your Arduino's pins to the motor drivers' pins:

Arduino Pin	VNH5019 Driver Pin	Basic Function
Digital 2	M1INA	Motor 1 direction input A
Digital 4	M1INB	Motor 1 direction input B
Digital 6	M1EN/DIAG	Motor 1 enable input/fault output
Digital 7	M2INA	Motor 2 direction input A
Digital 8	M2INB	Motor 2 direction input B
Digital 9	M1PWM	Motor 1 speed input
Digital 10	M2PWM	Motor 2 speed input
Digital 12	M2EN/DIAG	Motor 2 enable input/fault output
Analog 0	M1CS	Motor 1 current sense output
Analog 1	M2CS	Motor 2 current sense output

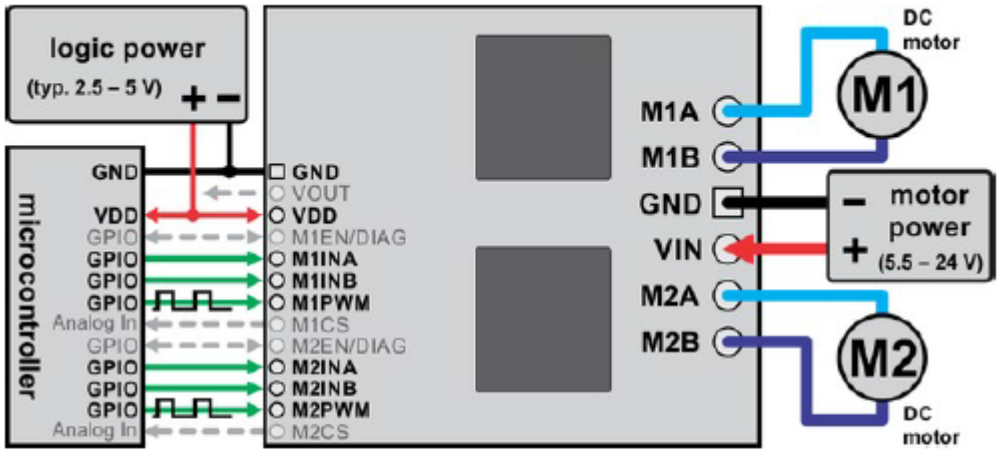
Power Connections and Considerations



Dual VNH5019 motor driver shield power buses when connected to an Arduino.

In the shield's default state, the motor driver shield and Arduino are powered separately. When used this way, the Arduino must be powered via USB, its power jack, or its VIN pin, and the shield must be supplied with 5.5 to 24 V through the large **VIN** and **GND** pads on the right side of the board. Attempting to power the shield through other means, such as from the Arduino or through the small VOUT pin, can permanently damage both the Arduino and the shield (only the large power traces on the right side of the shield are designed to handle the high currents involved in powering motors). A high-side reverse-voltage protection MOSFET prevents the shield from being damaged if shield power is inadvertently connected backwards. Logic power, VDD, is automatically supplied by the Arduino.

4.b. Board Connections

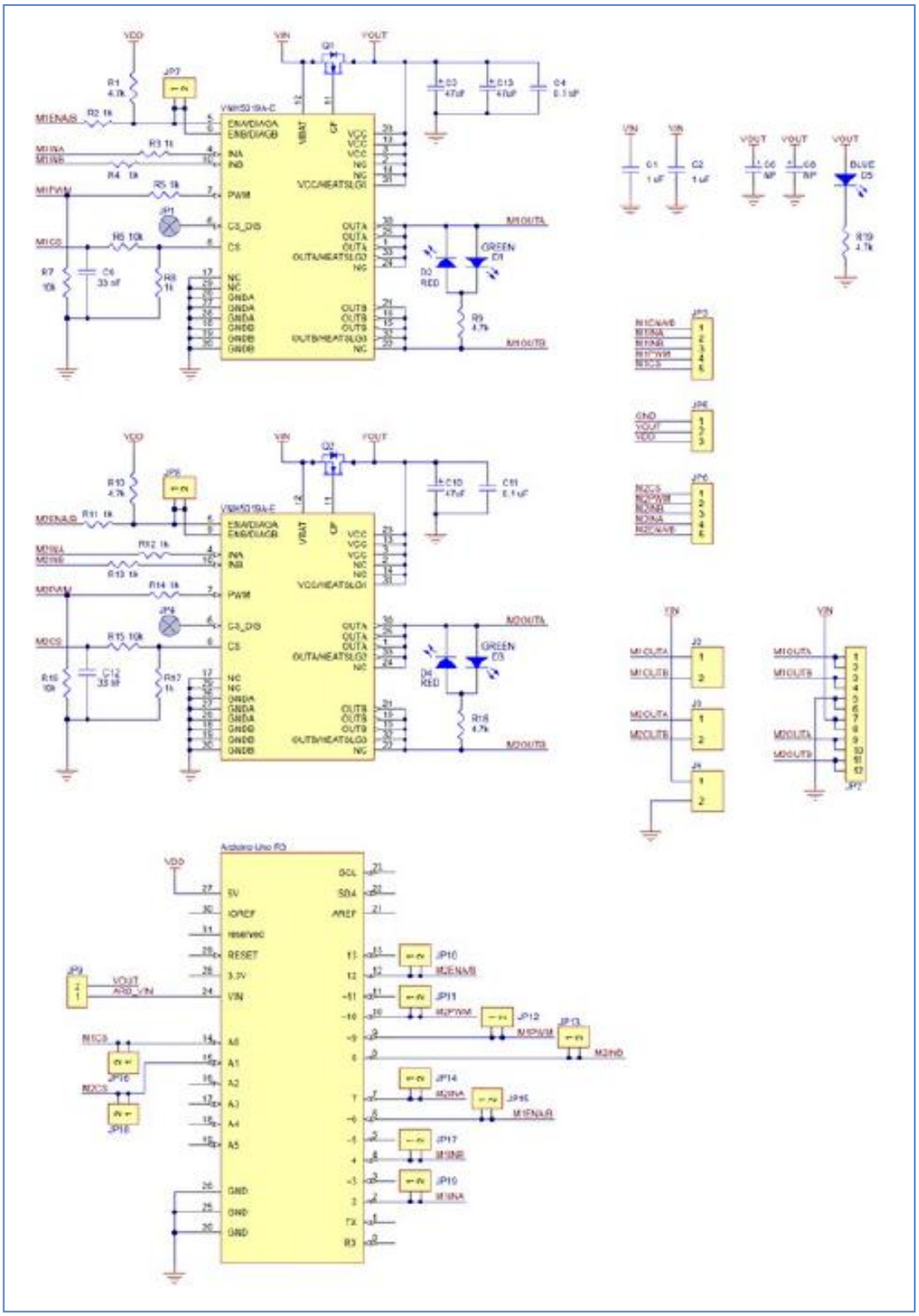


Dual VNH5019 motor driver shield connected to a microcontroller (gray connections are optional).

The above diagram shows the minimum connections typically required to interface this motor driver with a microcontroller.

Pinout

The following table explains the board pins in detail. See the **VNH5019 datasheet** [http://www.pololu.com/file/download/VNH5019A-E.pdf?file_id=0J504] (629k pdf) for even more detailed information about these pins, including the truth table that explains how the MxPWM and MxINA/B pins affect the MxA/B motor outputs.


VDD
VIN
VOUT
VNH5019A-E
M1ENA/B
M1INA
M1INB
M1PWM
M1CS
M1OUTA
M1OUTB
GREEN
RED
M2ENA/B
M2INA
M2INB
M2PWM
M2CS
M2OUTA
M2OUTB
Arduino Uno R3
GND
5V
3.3V
RESET
A0
A1
A2
A3
A4
A5
SCL
SDA
AREF
TX
RX

A.3 Metal Gear motor:

Gearmotor Dimensions

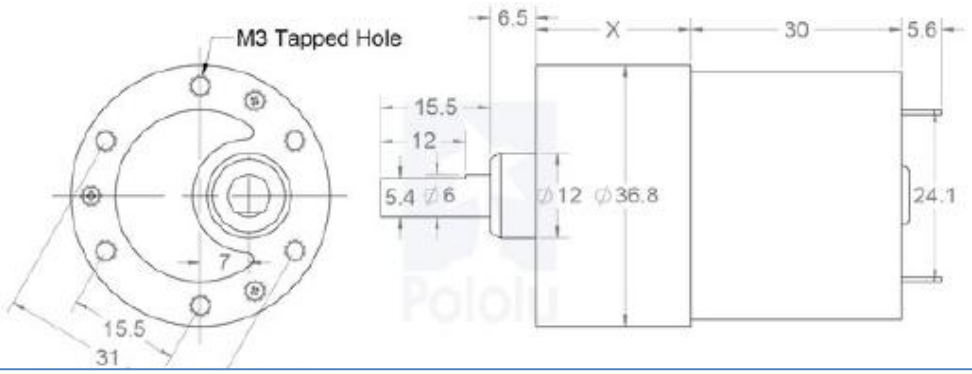
The face plate has six mounting holes evenly spaced around the outer edge threaded for M3 screws. These mounting holes form a regular hexagon and the centers of neighboring holes are 15.5 mm apart. You can use our custom 37D mm metal gearmotor bracket (shown in the left picture below) to mount the gearmotor to your project via these mounting holes and the screws that come with the bracket.

Gearmotor with bracket and hub.

37D mm metal gearmotor with 64 CPR encoder and Pololu 90×10mm wheel.

Please note that unlike our smaller metal gearmotors, these 37D mm gearmotors have output shafts with a diameter of 6 mm. The Pololu universal aluminum mounting hub for 6mm shafts can be used to mount our larger Pololu wheels (80mm- and 90mm-diameter) or custom wheels and mechanisms to the gearmotor's output shaft (see the right picture above).

The diagram below shows the dimensions (in mm) of the 37D mm line of gearmotors. The value of X is 22 mm for the 19:1 37Dx52L mm and 29:1 37Dx52L mm versions, 24 mm for the 50:1 37Dx54L mm and 67:1 37Dx54L mm versions, and 26.5 mm for the 100:1 37Dx57L mm and 131:1 37Dx57L mm versions. Note that the encoder PCB and magnetic disc are not shown in this dimension diagram. The encoder assembly extends an additional 12.5 mm beyond the rear of the motor.



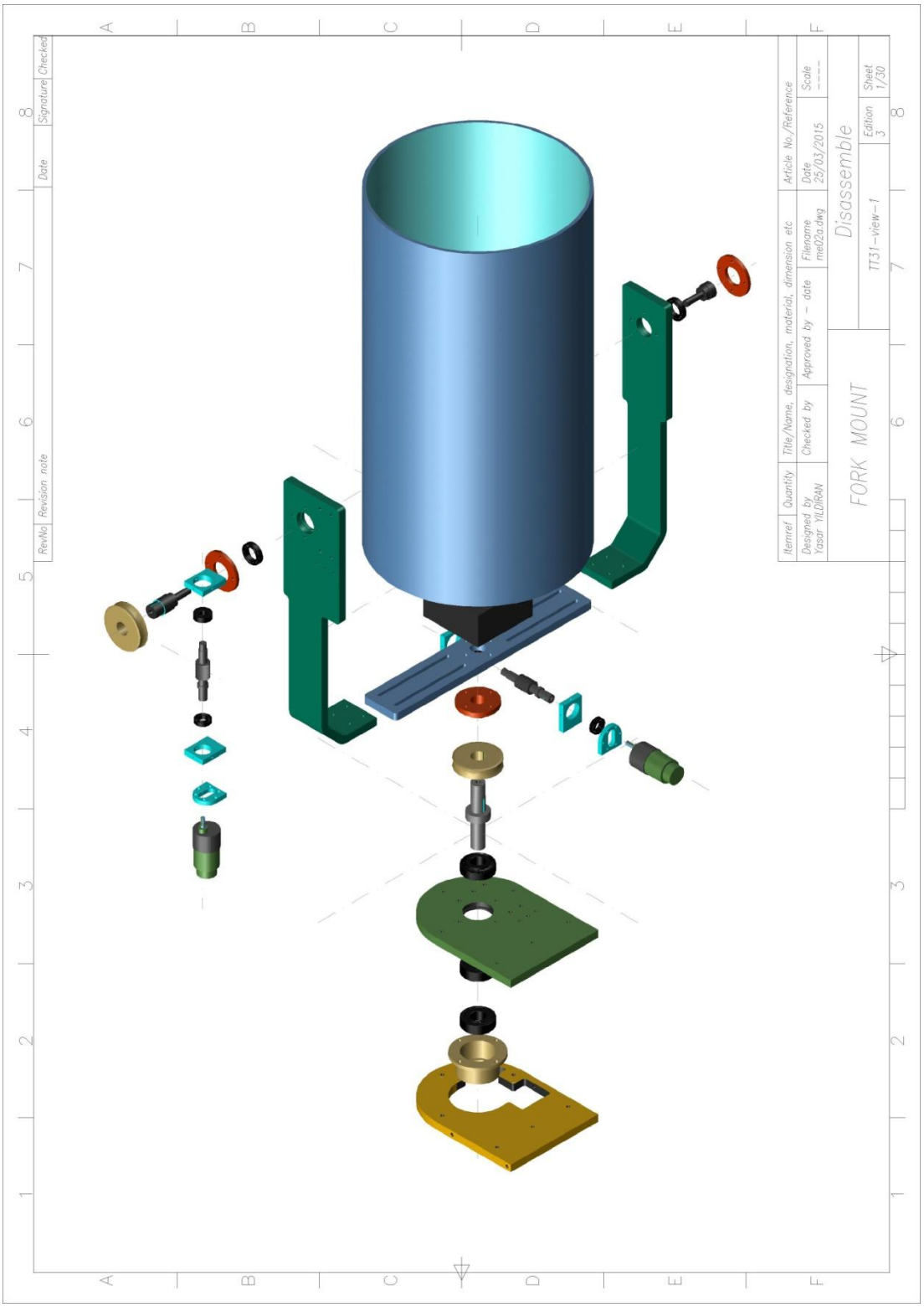
Using the Encoder

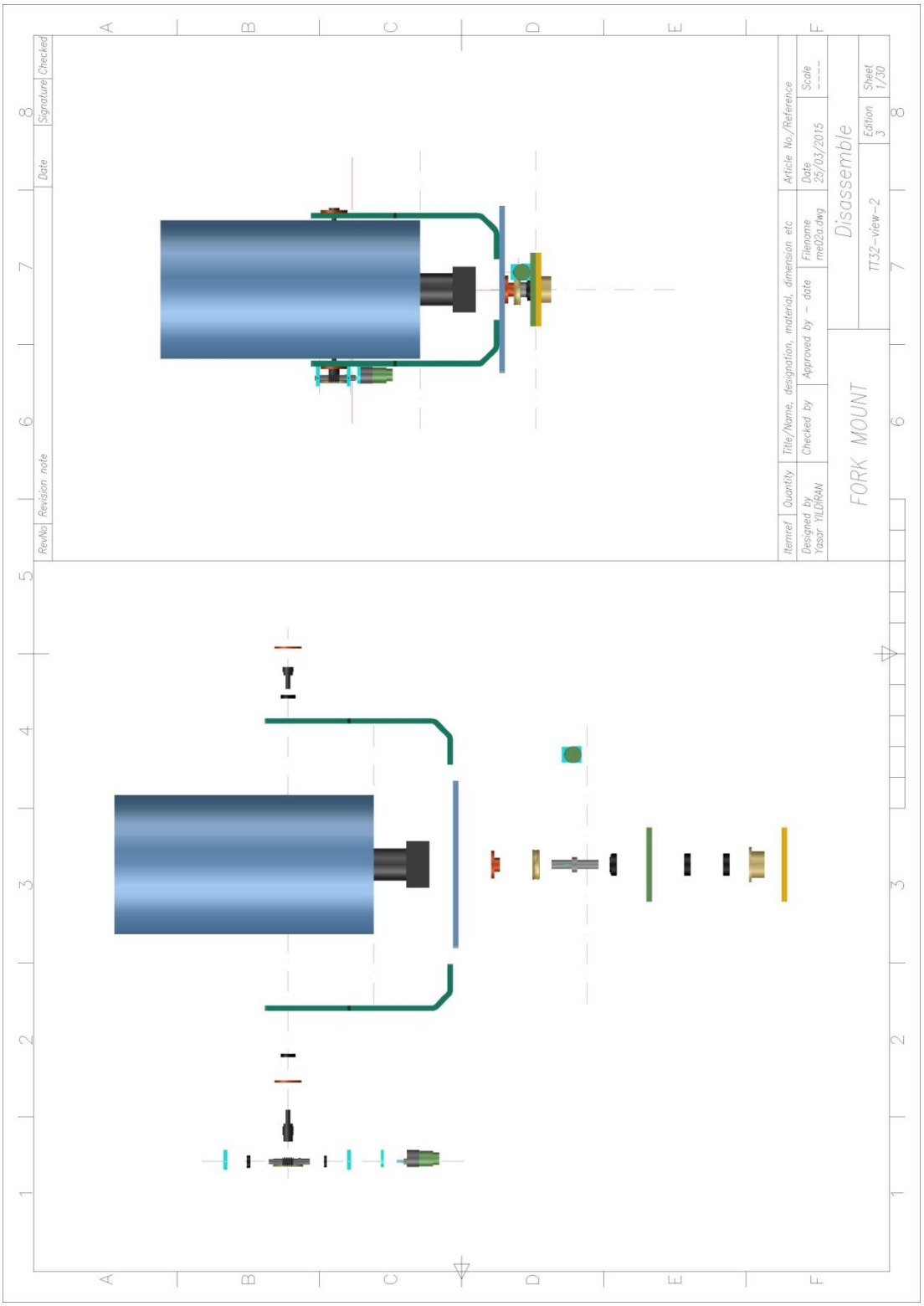
A two-channel Hall effect encoder is used to sense the rotation of a magnetic disk on a rear protrusion of the motor shaft. The quadrature encoder provides a resolution of 64 counts per revolution of the motor shaft. To compute the counts per revolution of the gearbox output, multiply the gear ratio by 64. The motor/encoder has six color-coded, 11" (28 cm) leads:

Color	Function
Black	motor power
Red	motor power
Blue	Hall sensor Vcc (3.5 – 20 V)
Green	Hall sensor GND
Yellow	Hall sensor A output
White	Hall sensor B output

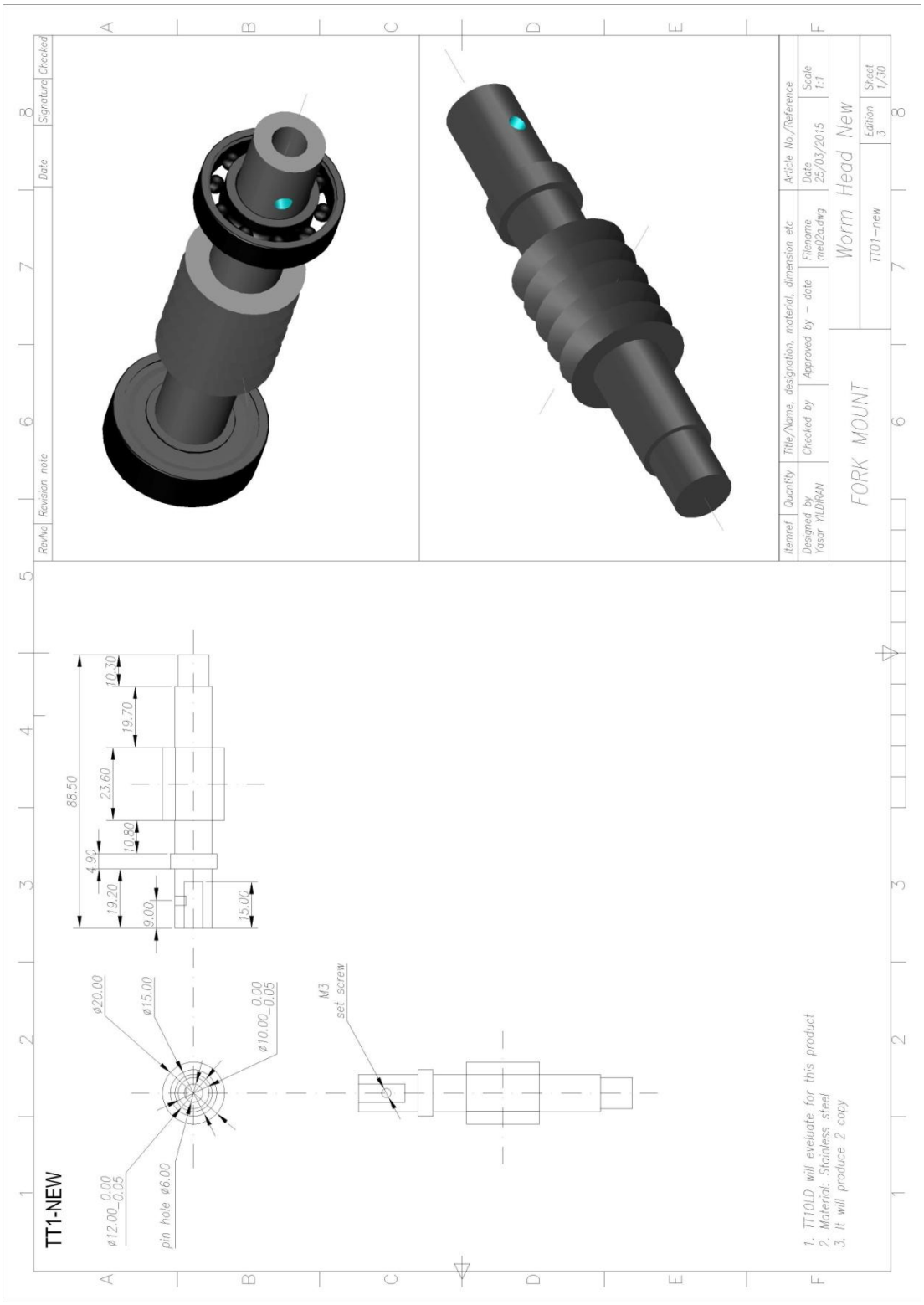
APPENDIX B

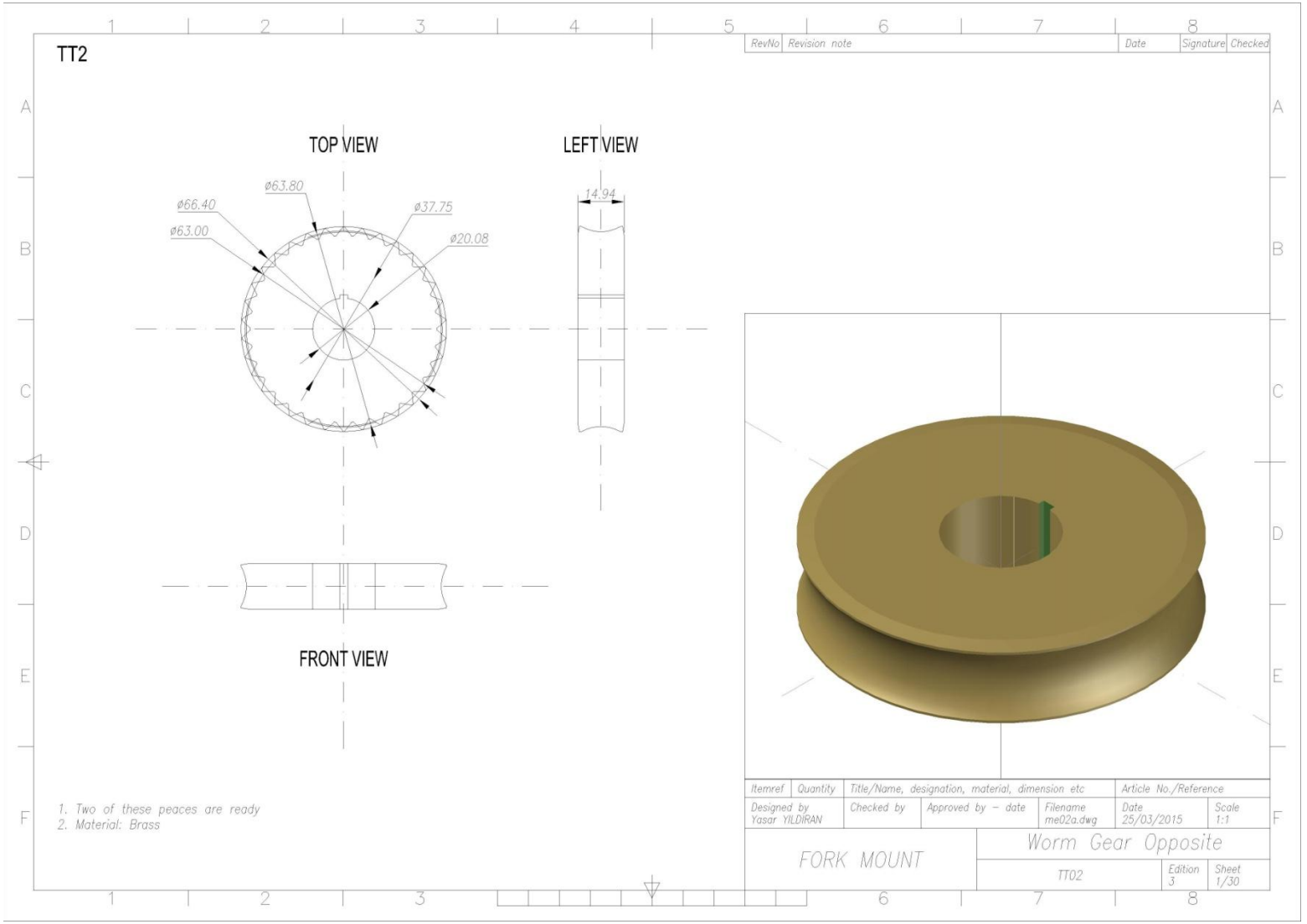
TELESCOPE MECHANICAL DESIGN

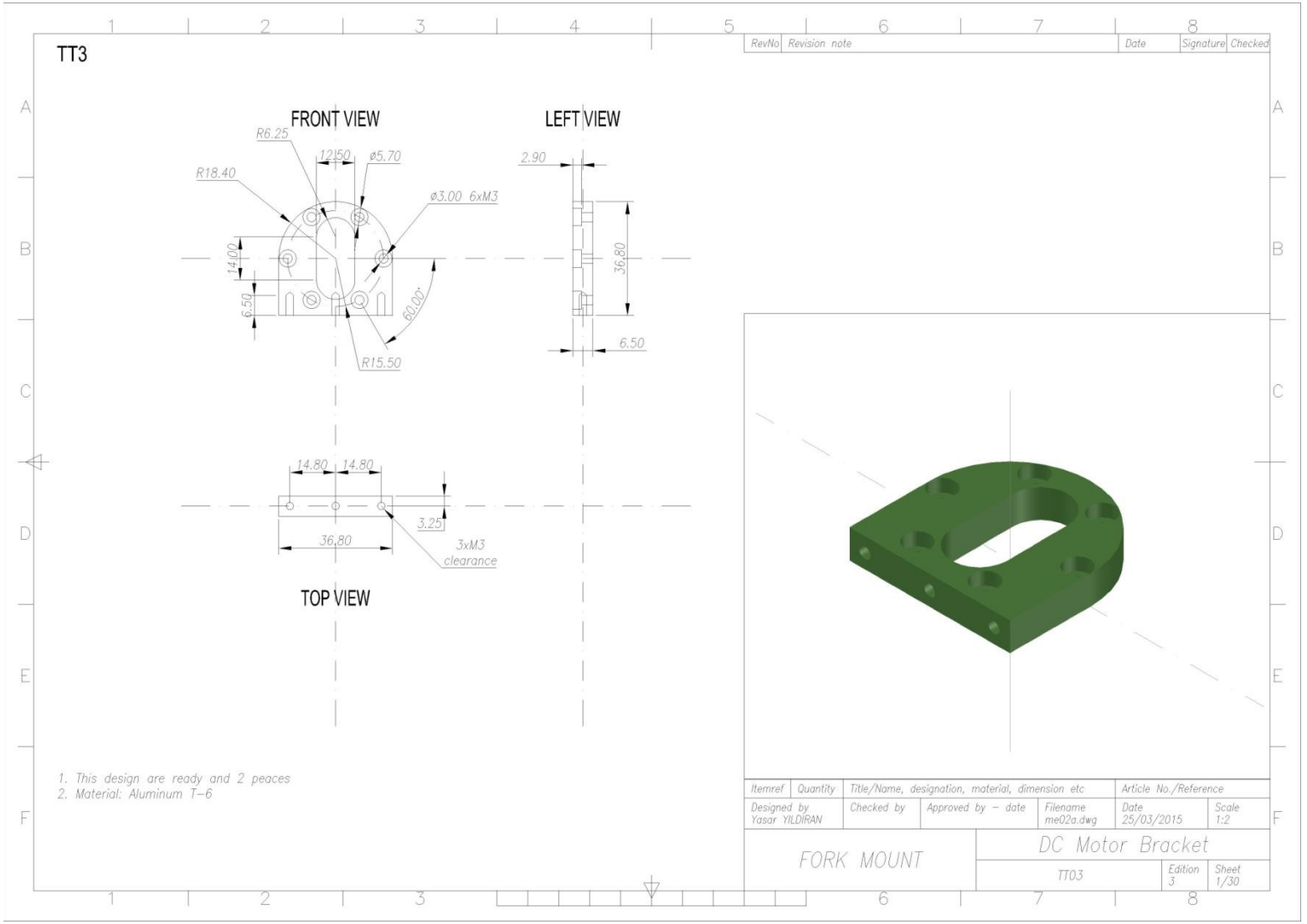



FORK MOUNT
Disassemble
TT32-view-2
Designed by
Yasar YILDIRAN
Checked by
Approved by - date
Filename
me02a.dwg
Date
25/03/2015
Scale

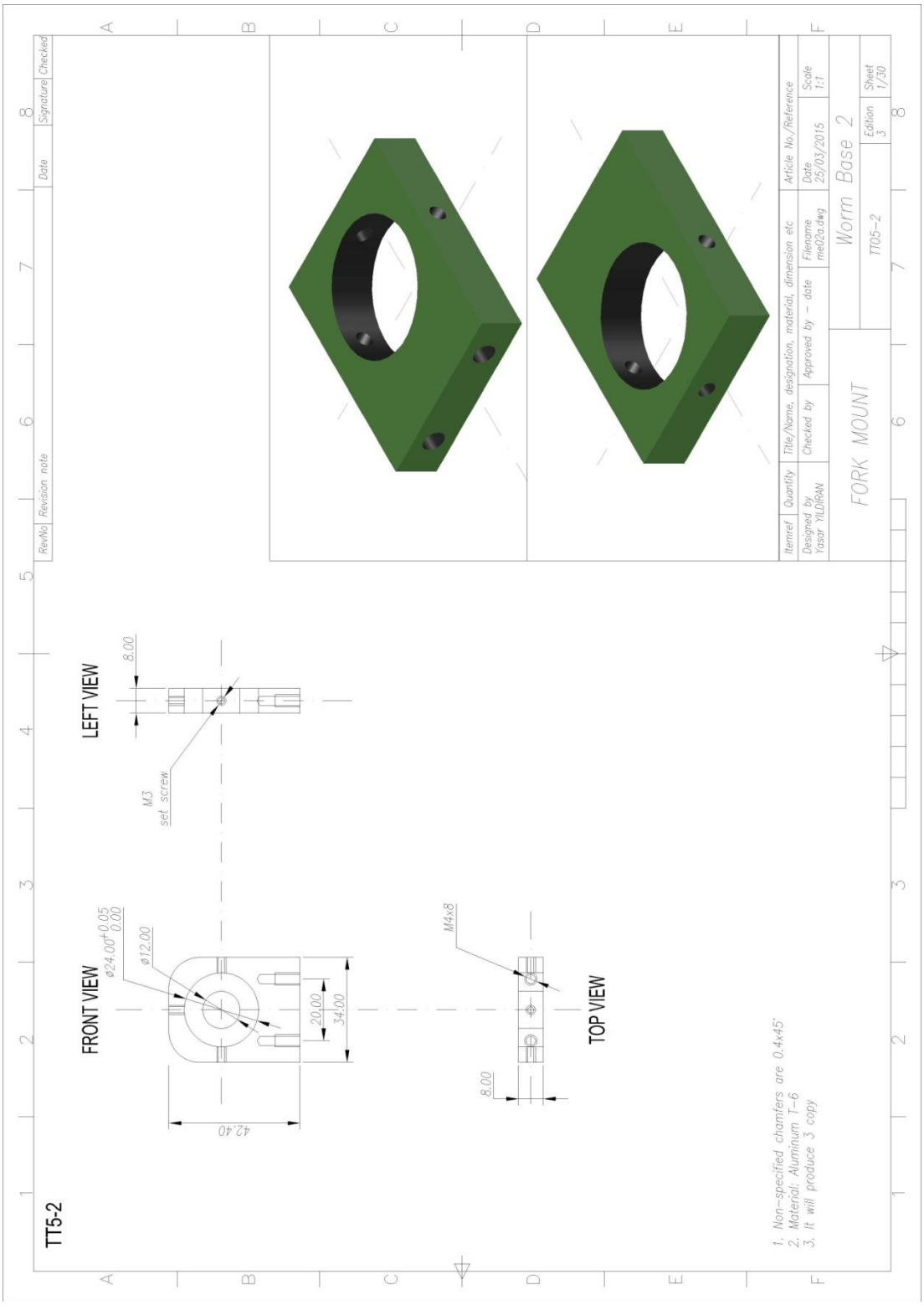
Edition
3
Sheet
1/30

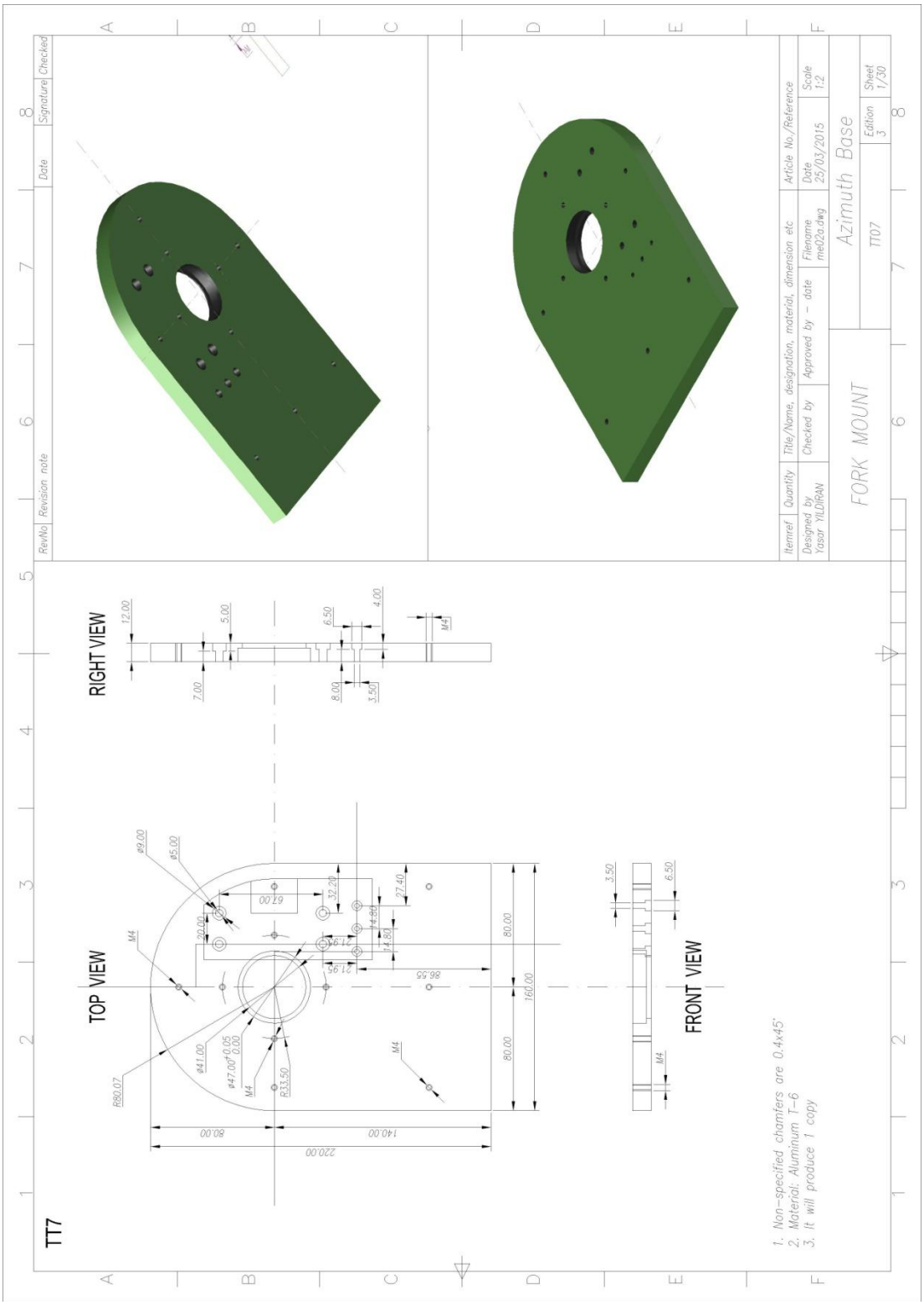

TT1-NEW
FORK MOUNT
Worm Head New
TT01-new
1. TT10LD will eveluate for this product
2. Material: Stainless steel
3. It will produce 2 copy
88.50
23.60
19.70
10.30
10.80
4.90
19.20
9.00
15.00
ø20.00
ø15.00
ø10.00 0.00 -0.05
ø12.00 0.00 -0.05
pin hole ø6.00
M3
set screw

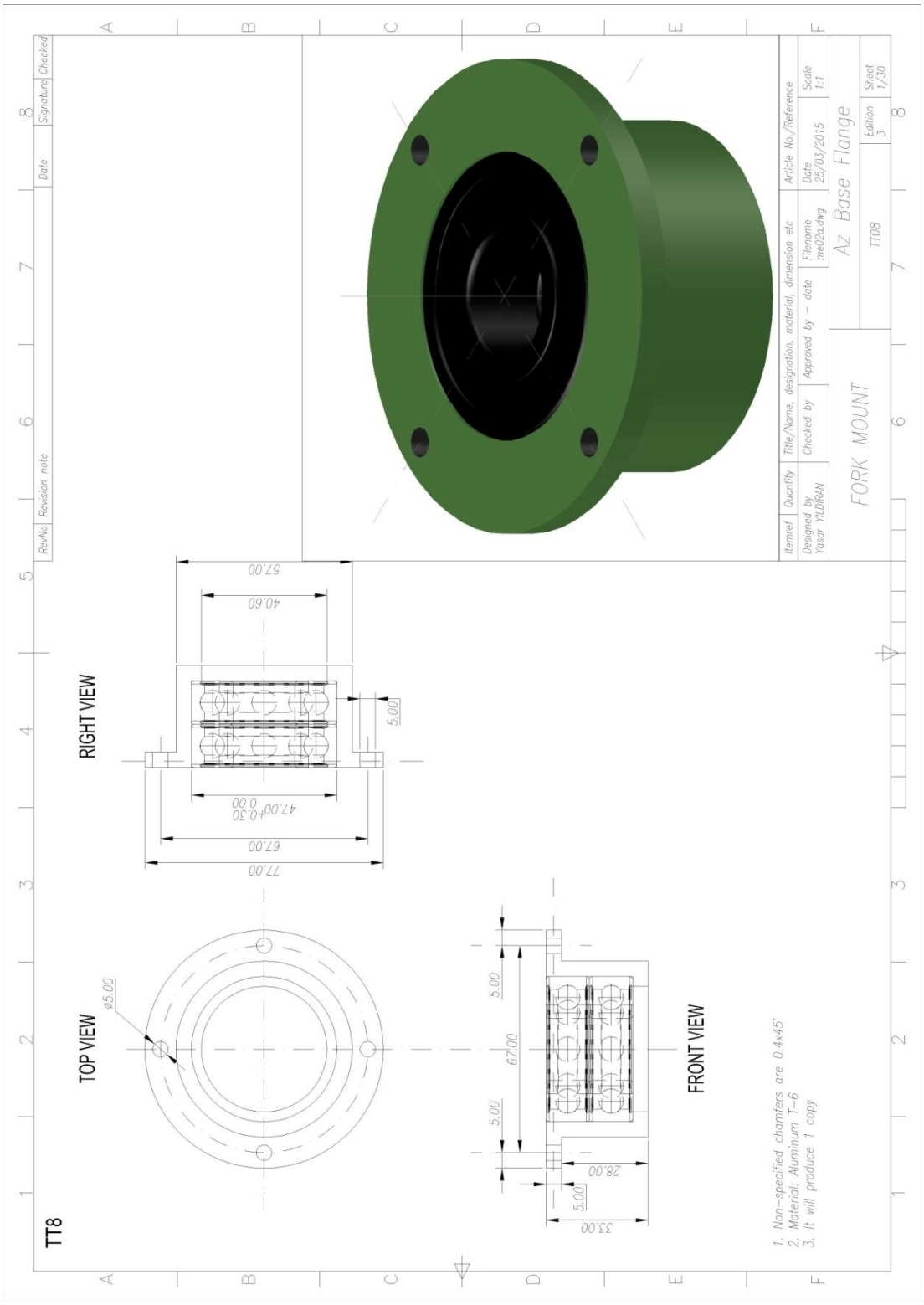

TT2
RevNo | Revision note | Date | Signature | Checked
TOP VIEW
LEFT VIEW
FRONT VIEW
ø63.80
ø66.40
ø63.00
ø37.75
ø20.08
14.94
1. Two of these peaces are ready
2. Material: Brass
Itemref | Quantity | Title/Name, designation, material, dimension etc | Article No./Reference
Designed by Yasar YILDIRAN | Checked by | Approved by – date | Filename me02a.dwg | Date 25/03/2015 | Scale 1:1
FORK MOUNT
Worm Gear Opposite
TT02 | Edition 3 | Sheet 1/30


TT3
RevNo Revision note Date Signature Checked
FRONT VIEW
LEFT VIEW
R6.25
12.50
ø5.70
R18.40
ø3.00 6xM3
14.00
6.50
60.00°
R15.50
2.90
36.80
6.50
14.80
14.80
3.25
36,80
3xM3
clearance
TOP VIEW
1. This design are ready and 2 peaces
2. Material: Aluminum T-6
Itemref Quantity Title/Name, designation, material, dimension etc Article No./Reference
Designed by Yasar YILDIRAN
Checked by
Approved by – date
Filename me02a.dwg
Date 25/03/2015
Scale 1:2
FORK MOUNT
DC Motor Bracket
TT03
Edition 3
Sheet 1/30

TT4
FRONT VIEW
RIGHT VIEW
TOP VIEW
ø12.00
ø36.80
6xM3
Tapped hole
ø6.00
7.00
0.60
5.40
12.00
15.50
6.50
26.50
30.00
13.20
27.00
34.50
36.80
1. This design are ready and 2 peaces
2. Material: Iron Cover
FORK MOUNT
DC motor
TT04
Edition 3
Sheet 1/30
Scale 1:1
Date 25/03/2015
Filename mel02a.dwg
Designed by Yasir YILDIRAN
Checked by
Approved by - date
Item/ref
Quantity
Title/Name, designation, material, dimension etc.
Article No./Reference
RevNo
Revision note
Date
Signature
Checked

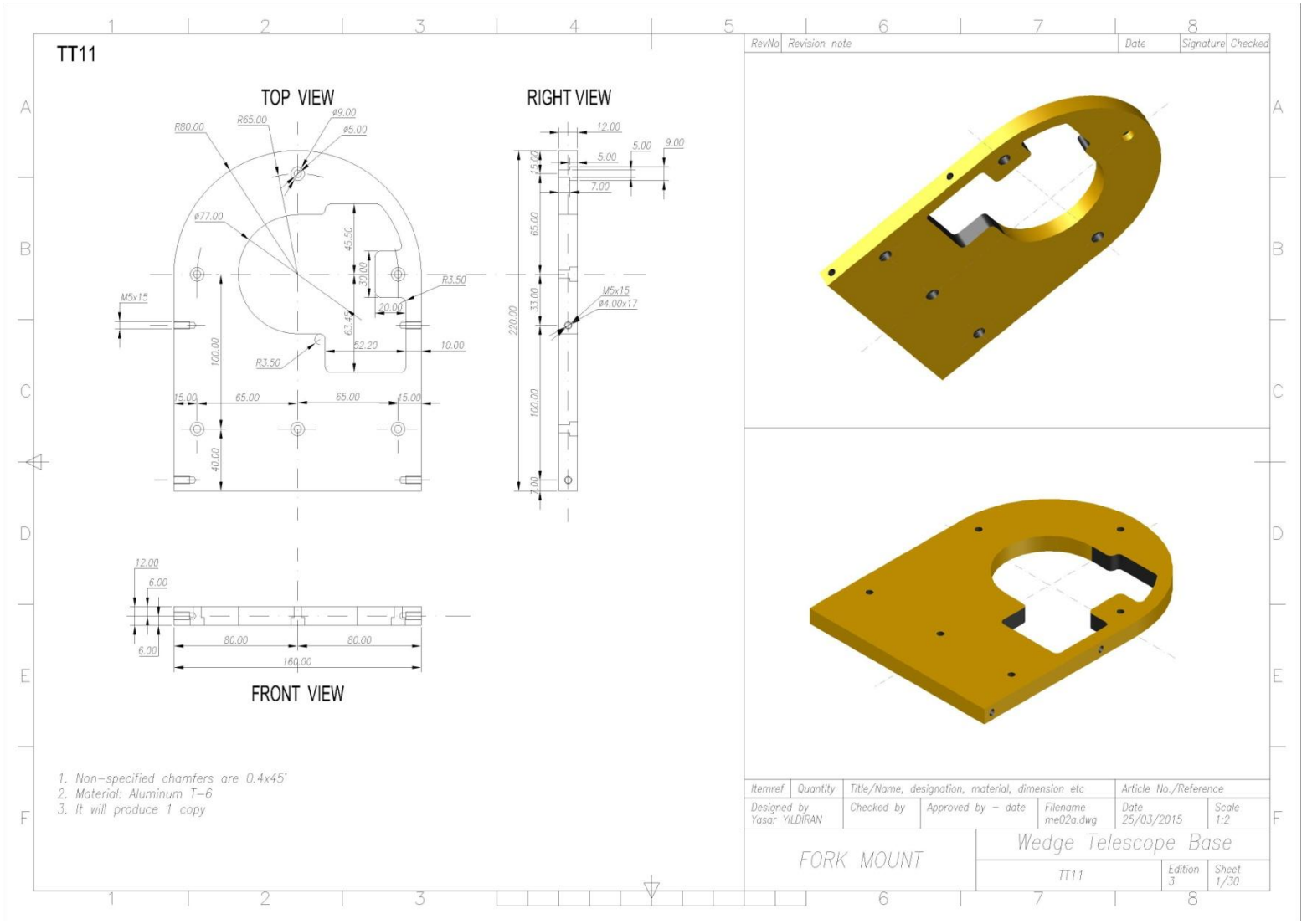

TT5-2
FRONT VIEW
LEFT VIEW
TOP VIEW
ø24.00 +0.05 0.00
ø12.00
42.40
20.00
34.00
8.00
M3
set screw
M4x8
8.00
1. Non-specified chamfers are 0.4x45°
2. Material: Aluminum T-6
3. It will produce 3 copy
RevNo. Revision note
Date
Signature Checked
Itemref. Quantity
Title/Name, designation, material, dimension etc.
Article No./Reference
Designed by Yasar YILDIRAN
Checked by
Approved by - date
Filename me02a.dwg
Date 25/03/2015
Scale 1:1
FORK MOUNT
TT05-2
Worm Base 2
Edition 3
Sheet 1/30

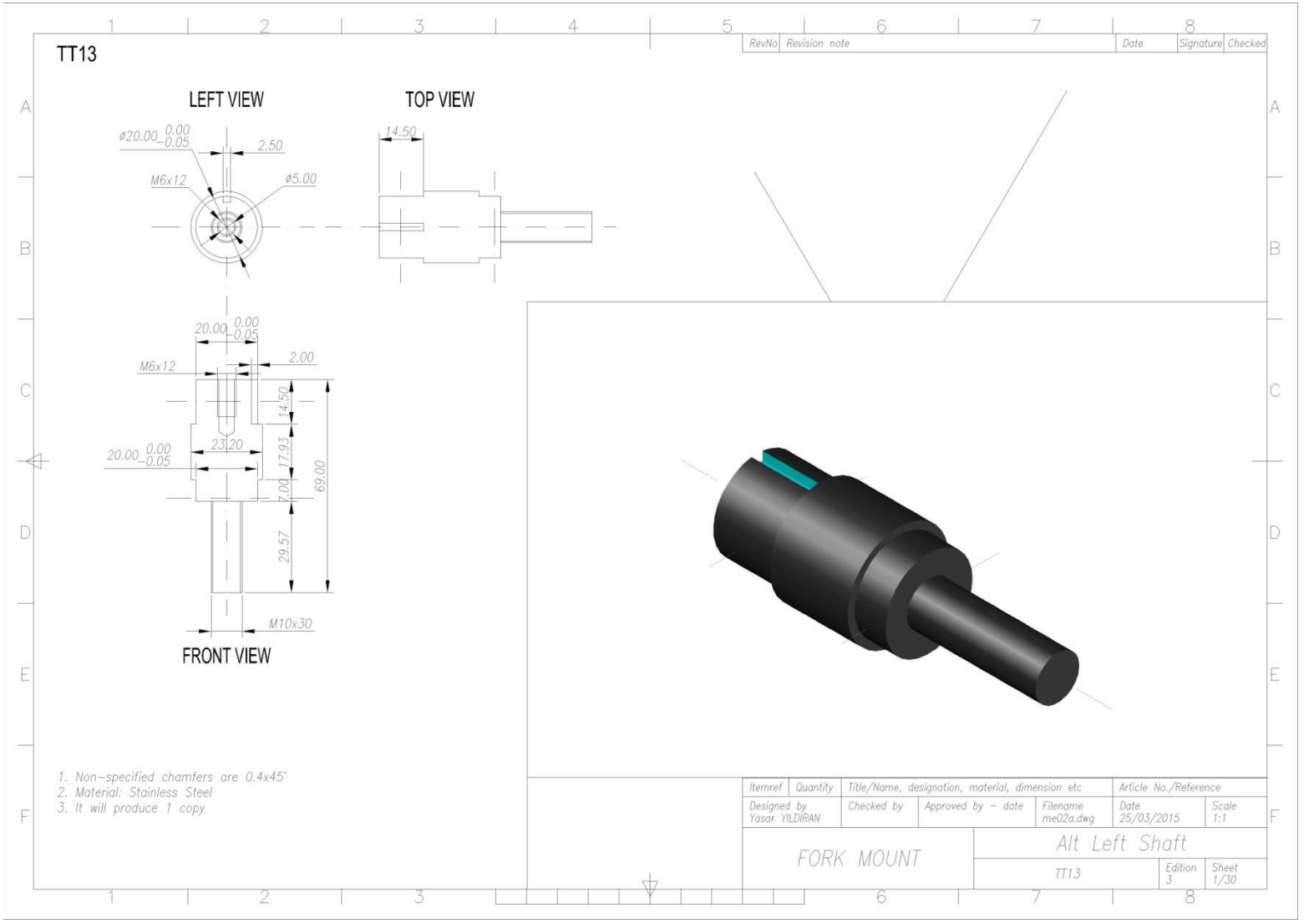

TT7
RIGHT VIEW
TOP VIEW
FRONT VIEW
FORK MOUNT
Azimuth Base
TT07
Designed by Yasir YILDIRAN
Checked by
Approved by - date
Filename mel02a.dwg
Date 25/03/2015
Scale 1:2
Edition 3
Sheet 1/30
1. Non-specified chamfers are 0.4x45°
2. Material: Aluminum T-6
3. It will produce 1 copy

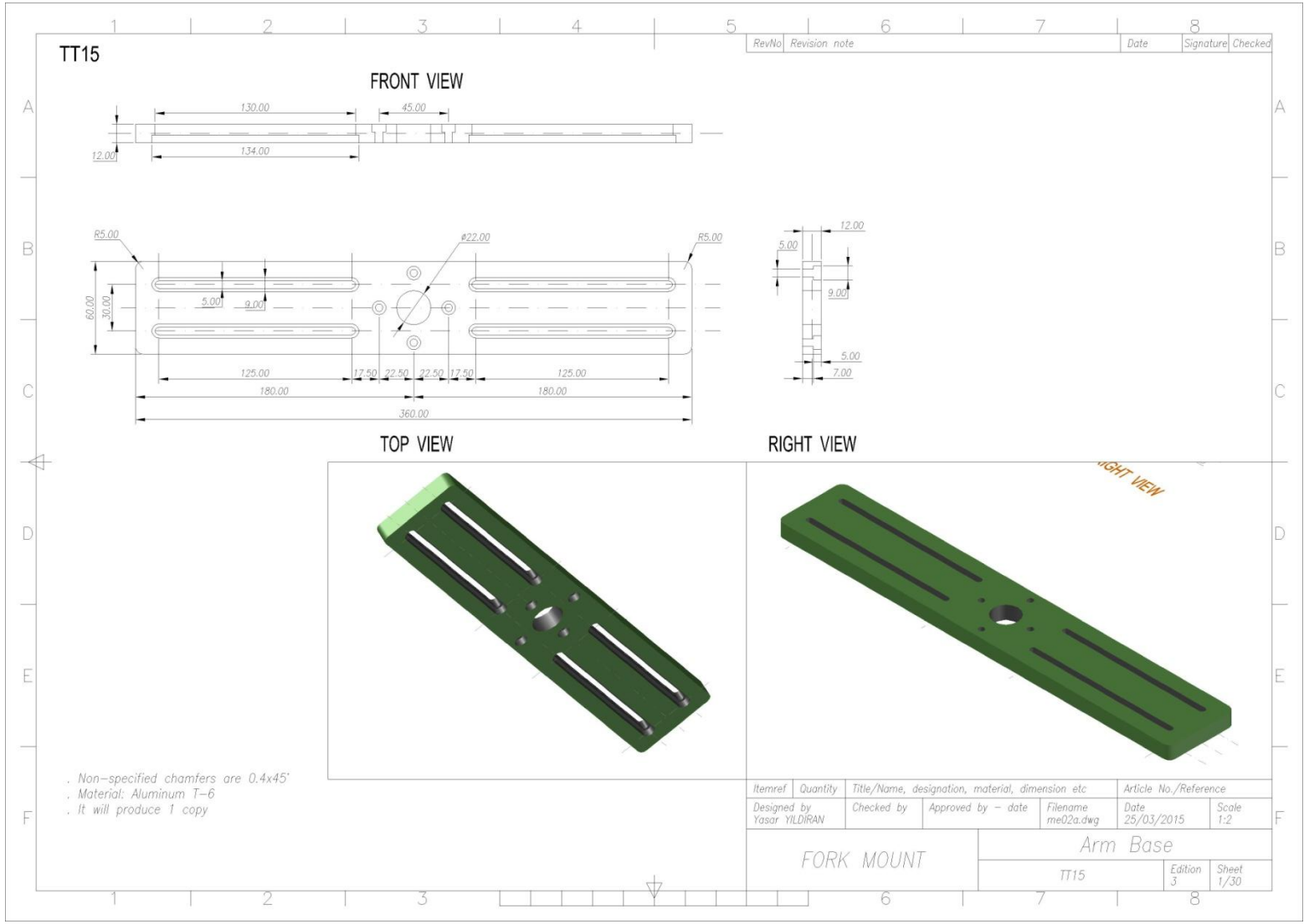

TT8
TOP VIEW
RIGHT VIEW
FRONT VIEW
ø5.00
57.00
40.60
5.00
47.00 +0.30 0.00
67.00
77.00
5.00
67.00
5.00
28.00
5.00
33.00
1. Non-specified chamfers are 0.4x45°
2. Material: Aluminum T-6
3. It will produce 1 copy
RevNo. Revision note
Date
Signature
Checked
Itemref. Quantity
Title/Name, designation, material, dimension etc.
Article No./Reference
Designed by Yasar YILDIRAN
Checked by
Approved by - date
Filename me02a.dwg
Date 25/03/2015
Scale 1:1
Edition 3
Sheet 1/30
FORK MOUNT
Az Base Flange
TT08

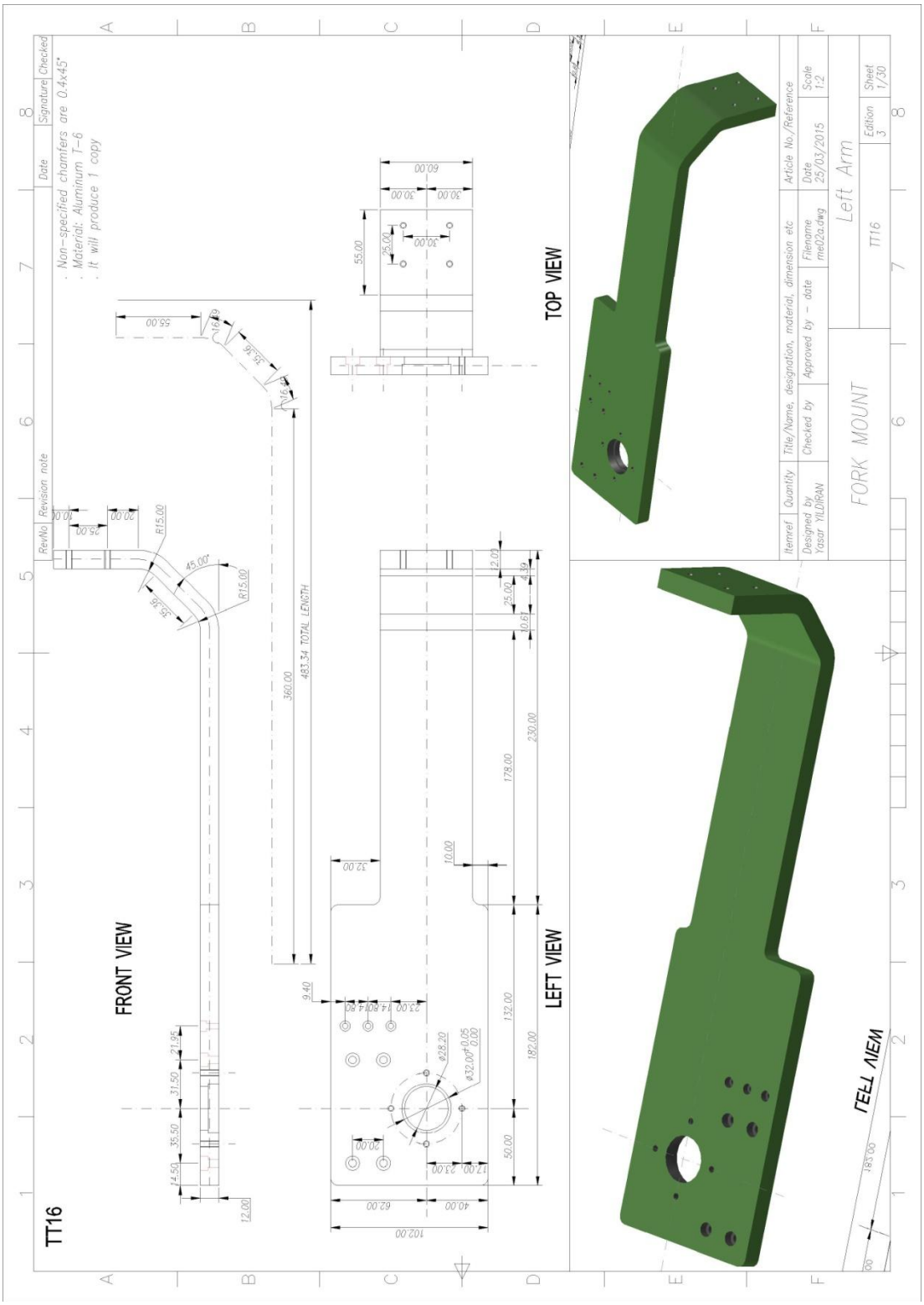
TT9
TOP VIEW
LEFT VIEW
FRONT VIEW
M6x12
M6x12
M10x16
46.25
43.57
101.50
11.68
22.00
5.00
2.50
2.00
7.00
15.00
101.50
Ø20.00 0.00 0.05
Ø33.20
1. Non-specified chamfers are 0.4x45°
2. Material: Stainless Steel
3. It will produce 1 copy
RevNo
Revision note
Date
Signature
Checked
Itemref
Quantity
Title/Name, designation, material, dimension etc.
Article No./Reference
Designed by
Yasar YILDIRAN
Checked by
Approved by - date
Filename
me02a.dwg
Date
25/03/2015
Scale
1:1
FORK MOUNT
Az Shaft
TT09
Edition
3
Sheet
1/30

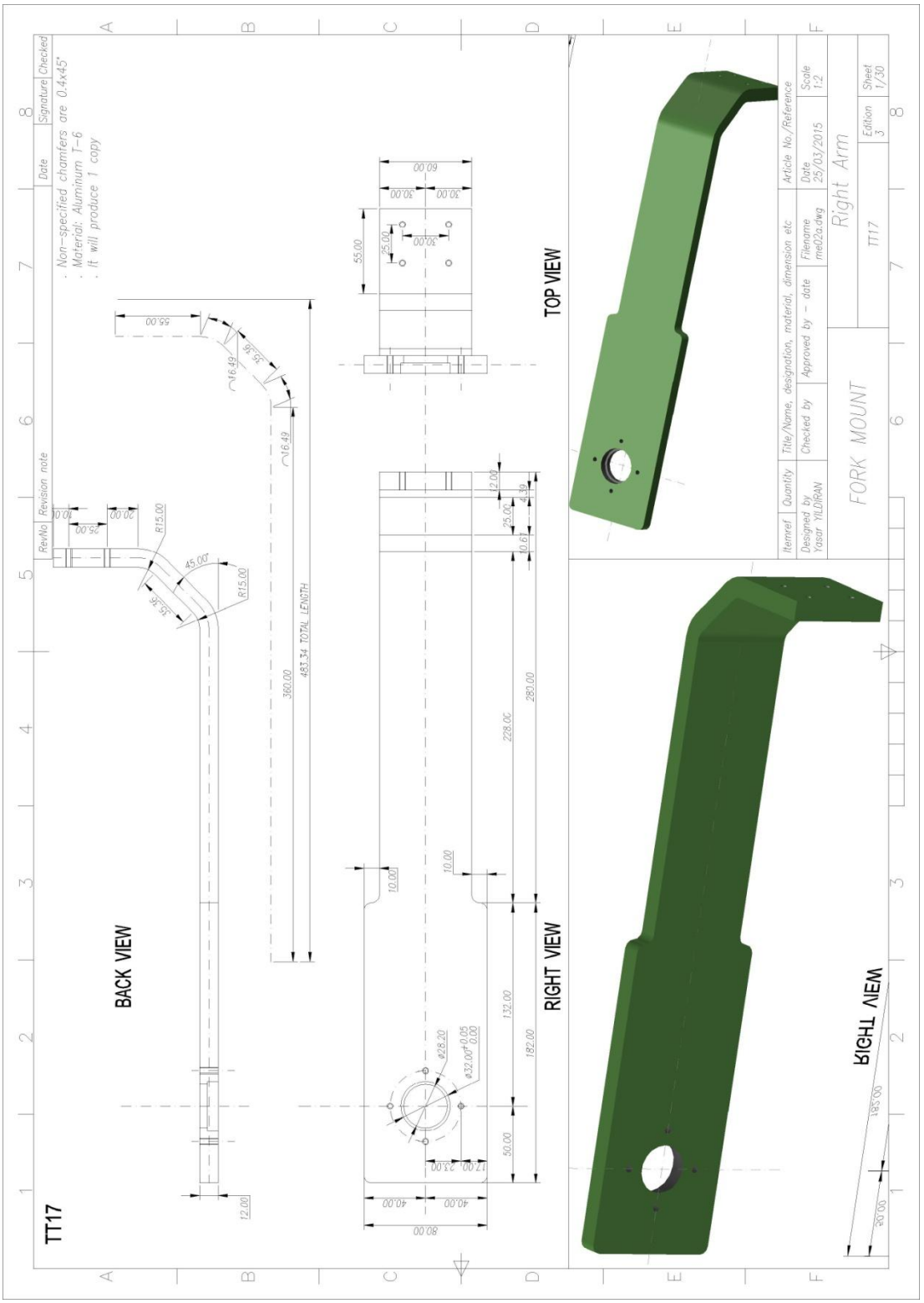
TT10
TOP VIEW
RIGHT VIEW
FRONT VIEW
M4
45.00°
M4
ø60.00
ø30.00
ø20.00 +0.10 0.00
ø45.00
M4
M3 set screw
7.00
6.00
19.13
12.12
1. Non-specified chamfers are 0.4x45°
2. Material: Aluminum T-6
3. It will produce 1 copy
RevNo
Revision note
Date
Signature
Checked
Itemref
Quantity
Title/Name, designation, material, dimension etc.
Article No./Reference
Designed by
Yasir YILDIRAN
Checked by
Approved by - date
Filename
me02a.dwg
Date
25/03/2015
Scale
1:1
FORK MOUNT
Arm Base Flange
TT10
Edition
3
Sheet
1/30


TT11
TOP VIEW
RIGHT VIEW
FRONT VIEW
R80.00
R65.00
ø9.00
ø5.00
ø77.00
45.50
30.00
R3.50
20.00
M5x15
63.45
52.20
10.00
R3.50
100.00
15.00
65.00
65.00
15.00
40.00
12.00
5.00
9.00
5.00
7.00
15.00
65.00
33.00
M5x15
ø4.00x17
220.00
100.00
7.00
12.00
6.00
6.00
80.00
80.00
160.00
1. Non-specified chamfers are 0.4x45°
2. Material: Aluminum T-6
3. It will produce 1 copy
RevNo
Revision note
Date
Signature
Checked
Itemref
Quantity
Title/Name, designation, material, dimension etc
Article No./Reference
Designed by
Yasar YILDIRAN
Checked by
Approved by - date
Filename
me02a.dwg
Date
25/03/2015
Scale
1:2
FORK MOUNT
Wedge Telescope Base
TT11
Edition
3
Sheet
1/30


TT13
RevNo Revision note Date Signature Checked
LEFT VIEW
TOP VIEW
ø20.00 0.00 −0.05
2.50
M6x12
ø5.00
14.50
20.00 0.00 −0.05
2.00
M6x12
14.50
23.20
20.00 0.00 −0.05
17.93
7.00
69.00
29.57
M10x30
FRONT VIEW
1. Non−specified chamfers are 0.4x45°
2. Material: Stainless Steel
3. It will produce 1 copy
Itemref Quantity Title/Name, designation, material, dimension etc Article No./Reference
Designed by Yasar YILDIRAN
Checked by
Approved by − date
Filename me02a.dwg
Date 25/03/2015
Scale 1:1
FORK MOUNT
Alt Left Shaft
TT13
Edition 3
Sheet 1/30


TT15
RevNo Revision note Date Signature Checked
FRONT VIEW
130.00
45.00
134.00
12.00
R5.00
ø22.00
R5.00
60.00
30.00
5.00
9.00
125.00
17.50
22.50
22.50
17.50
125.00
180.00
180.00
360.00
12.00
5.00
9.00
5.00
7.00
TOP VIEW
RIGHT VIEW
. Non-specified chamfers are 0.4x45°
. Material: Aluminum T-6
. It will produce 1 copy
Itemref Quantity Title/Name, designation, material, dimension etc Article No./Reference
Designed by Yasar YILDIRAN
Checked by
Approved by - date
Filename me02a.dwg
Date 25/03/2015
Scale 1:2
FORK MOUNT
Arm Base
TT15
Edition 3
Sheet 1/30


TT16
FRONT VIEW
LEFT VIEW
TOP VIEW
LEFT VIEW
. Non-specified chamfers are 0.4x45°
. Material: Aluminum T-6
. It will produce 1 copy
RevNo | Revision note | Date | Signature | Checked
Itemref | Quantity | Title/Name, designation, material, dimension etc. | Article No./Reference
Designed by Yasar YILDIRAN | Checked by | Approved by - date | Filename me02a.dwg | Date 25/03/2015 | Scale 1:2
FORK MOUNT
TT16
Left Arm
Edition 3
Sheet 1/30
483.34 TOTAL LENGTH
360.00
178.00
230.00
132.00
182.00
102.00
62.00
40.00
50.00
55.00
R15.00
45.00°
ø28.20
ø32.00 +0.05 0.00


TT17
BACK VIEW
TOP VIEW
RIGHT VIEW
RIGHT VIEW
FORK MOUNT
Right Arm
TT17
Edition 3
Sheet 1/30
Scale 1:2
Date 25/03/2015
Filename me02a.dwg
Designed by Yasar YILDIRAN
Checked by
Approved by - date
Article No./Reference
Title/Name, designation, material, dimension etc
Quantity
Itemref
RevNo
Revision note
Date
Signature
Checked
. Non-specified chamfers are 0.4x45°
. Material: Aluminum T-6
. It will produce 1 copy
483.34 TOTAL LENGTH
360.00
280.00
228.06
182.00
132.00
80.00
60.00
55.00
45.00
30.00
25.00
20.00
15.00
12.00
10.00
4.39
0.6
R15.00
R16.49
Ø28.20
Ø32.00 +0.05 0.00

APPENDIX C

ENCODER READ C CODE

```
# include <Arduino.h>

typedef struct { int pinA; int pinB; int pos; int del;} Encoder;
volatile Encoder Enc[3] = {{0,0,0,0}, {0,0,0,0}, {0,0,0,0}};

/* auxiliary function to handle encoder attachment           */
int getIntNum(int pin) {
/* returns the interrupt number for a given interrupt pin
   see http://arduino.cc/it/Reference/AttachInterrupt        */
switch(pin) {
    case 21:
    return 2;
  case 20:
    return 3;
  case 19:
    return 4;
  case 18:
    return 5;
  default:
    return -1;
  }
}

/* auxiliary debouncing function                             */
void debounce(int del) {
  for (int k=0;k<del;k++) {
    /* can't use delay in the ISR so need to waste some time
       perfoming operations, this uses roughly 0.1ms on uno  */
    k = k +0.0 +0.0 -0.0 +3.0 -3.0;
  }
}

/* Interrupt Service Routine: change on pin A for Encoder 1  */
void irsPinAEn1(){

  /* read pin B right away                                   */
  int drB = digitalRead(Enc[1].pinB);

  /* possibly wait before reading pin A, then read it        */
  debounce(Enc[1].del);
  int drA = digitalRead(Enc[1].pinA);

  /* this updates the counter                                */
  if (drA == HIGH) {   /* low->high on A? */

    if (drB == LOW) {  /* check pin B */
    Enc[1].pos++;  /* going clockwise: increment         */
    } else {
    Enc[1].pos--;  /* going counterclockwise: decrement  */
    }

  } else { /* must be high to low on A                       */

    if (drB == HIGH) { /* check pin B */
```

```
    Enc[1].pos++;  /* going clockwise: increment         */
    } else {
    Enc[1].pos--;  /* going counterclockwise: decrement  */
    }

  } /* end counter update                                 */

} /* end ISR pin A Encoder 1                             */


/* Interrupt Service Routine: change on pin B for Encoder 1  */
void isrPinBEn1(){

  /* read pin A right away                                 */
  int drA = digitalRead(Enc[1].pinA);

  /* possibly wait before reading pin B, then read it      */
  debounce(Enc[1].del);
  int drB = digitalRead(Enc[1].pinB);

  /* this updates the counter                              */
  if (drB == HIGH) {   /* low->high on B? */

    if (drA == HIGH) { /* check pin A */
    Enc[1].pos++;  /* going clockwise: increment         */
    } else {
    Enc[1].pos--;  /* going counterclockwise: decrement  */
    }

  } else { /* must be high to low on B                    */

    if (drA == LOW) {  /* check pin A */
    Enc[1].pos++;  /* going clockwise: increment         */
    } else {
    Enc[1].pos--;  /* going counterclockwise: decrement  */
    }

  } /* end counter update                                 */

} /* end ISR pin B Encoder 1                             */


/* Interrupt Service Routine: change on pin A for Encoder 2  */
void irsPinAEn2(){

  /* read pin B right away                                 */
  int drB = digitalRead(Enc[2].pinB);

  /* possibly wait before reading pin A, then read it      */
  debounce(Enc[2].del);
  int drA = digitalRead(Enc[2].pinA);

  /* this updates the counter                              */
  if (drA == HIGH) {   /* low->high on A? */

    if (drB == LOW) {  /* check pin B */
    Enc[2].pos++;  /* going clockwise: increment         */
    } else {
```

```
    Enc[2].pos--;  /* going counterclockwise: decrement  */
    }

  } else { /* must be high to low on A                        */

    if (drB == HIGH) { /* check pin B */
    Enc[2].pos++;  /* going clockwise: increment         */
    } else {
    Enc[2].pos--;  /* going counterclockwise: decrement  */
    }

  } /* end counter update                                    */

} /* end ISR pin A Encoder 2                                 */


/* Interrupt Service Routine: change on pin B for Encoder 2  */
void isrPinBEn2(){

  /* read pin A right away                                   */
  int drA = digitalRead(Enc[2].pinA);

  /* possibly wait before reading pin B, then read it        */
  debounce(Enc[2].del);
  int drB = digitalRead(Enc[2].pinB);

  /* this updates the counter                                */
  if (drB == HIGH) {   /* low->high on B? */

    if (drA == HIGH) { /* check pin A */
    Enc[2].pos++;  /* going clockwise: increment         */
    } else {
    Enc[2].pos--;  /* going counterclockwise: decrement  */
    }

  } else { /* must be high to low on B                        */

    if (drA == LOW) {  /* check pin A */
    Enc[2].pos++;  /* going clockwise: increment         */
    } else {
    Enc[2].pos--;  /* going counterclockwise: decrement  */
    }

  } /* end counter update                                    */

} /* end ISR pin B Encoder 2                                 */


void enc_init(int enc, int pinA, int pinB) {

        /* enc is the encoder number and it can be 1 or 2        */
        /* if other encoder blocks are present in the model      */
        /* up to a maximum of 3, they need to refer to a         */
        /* different encoder number                              */

        /* store pinA and pinB in global encoder structure Enc   */
        /* they will be needed later by the interrupt routine    */
        /* that will not be able to access s-function parameters */
```

```
        Enc[enc].pinA=pinA;      /* set pin A              */
        Enc[enc].pinB=pinB;      /* set pin B              */

        /* set encoder pins as inputs                              */
        pinMode(Enc[enc].pinA, INPUT);
        pinMode(Enc[enc].pinB, INPUT);

        /* attach interrupts                                       */
        switch(enc) {

            case 1:
            attachInterrupt(getIntNum(Enc[1].pinA), irsPinAEn1,
CHANGE);
            attachInterrupt(getIntNum(Enc[1].pinB), isrPinBEn1,
CHANGE);
            break;
          case 2:
            attachInterrupt(getIntNum(Enc[2].pinA), irsPinAEn2,
CHANGE);
            attachInterrupt(getIntNum(Enc[2].pinB), isrPinBEn2,
CHANGE);
            break;
          }
}

int enc_output(int enc) {
    return (double)Enc[enc].pos;
}
```